

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему:

Розробка вебплатформи для онлайн замовлення
та доставки харчових продуктів

Виконав: студент
спеціальності

IV курсу, групи СН-42
122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Максимчак А.В.
(прізвище та ініціали)

Керівник

(підпис)

Гром'як Р. С..
(прізвище та ініціали)

Нормоконтроль

(підпис)

Липак Г.І.
(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.
(прізвище та ініціали)

Рецензент

(підпис)

Луцик Н.С.
(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 8 » 06 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Максимчуку Антону Васильовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебплатформи для онлайн замовлення та доставки харчових продуктів

Керівник роботи Гром'як Роман Сильвестрович к. ф. – м. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 14 » травня 2026 року № 4/7-239

2. Термін подання студентом завершеної роботи 23 червня 2026р.

3. Вихідні дані до роботи Літературні та інтернет джерела інформації по технології розробки програмного комплексу для розробки веб платформи

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Постановка задачі та формування вимог до програмного комплексу. 1.2. Аналіз предметної області. 1.2. Формування вимог до програмного комплексу. 1.3. Пошук актантів та варіантів використання. 1.4. Вибір середовища розробки. 1.5. Обґрунтування використовуваних технологій. 1.6. Висновок до першого розділу. 2. Проектування, реалізація та тестування програмного комплексу. 2.1. Формування структури програмного комплексу. 2.2. Проектування та створення БД для розроблюваного комплексу. 2.3. Проектування інтерфейсу користувача. 2.4. Розробка програмного комплексу. 2.4.1. Розробка інтерфейсу. 2.4.2. Розробка функцій для обробки даних. 2.5. Тестування розробленого програмного комплексу. 2.6. Перспективи модернізації. 2.7. Висновок до другого розділу. 3. Безпека життєдіяльності, основи охорони праці. 3.1. Працездатність людини-оператора. 3.2. Загальні вимоги безпеки з охорони праці для користувачів ПК. 3.3. Висновок до третього розділу. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Мета кваліфікаційної роботи. 3. Актуальність обраної теми.

4. Аналіз предметної області. 5. Актори програмного комплексу. 6. Використані технології.

7. Структура програмного комплексу. 8. База даних. 9. Вигляд користувацької частини сайту.

10. Вигляд панелі адміністратора. 11. Висновки.

АНОТАЦІЯ

Розробка вебплатформи для онлайн замовлення та доставки харчових продуктів
// Кваліфікаційна робота освітнього рівня «Бакалавр» //Максимчак Антон //
Тернопільський національний технічний університет імені Івана Пулюя,
факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра
комп'ютерних наук, група СН-42 // Тернопіль, 2026 // С. 60, рис. – 15 , додат. –
3, бібліогр. – 46.

Ключові слова: сайт, програмний комплекс, товари, react, node, express.js.

Кваліфікаційна робота присвячена розробці вебплатформи для онлайн замовлення та доставки харчових продуктів.

У першому розділі кваліфікаційної роботи описано постановку задачі та формування вимог до вебплатформи. Проаналізовано предметну область дослідження, визначено основних користувачів системи та варіанти використання. Розглянуто існуючі аналоги систем доставки продуктів, а також обґрунтовано вибір технологій для реалізації проєкту.

У другому розділі кваліфікаційної роботи сформовано загальну структуру вебплатформи, спроектовано та створено базу даних. Розроблено користувацький інтерфейс та адміністративну панель, реалізовано основні функції обробки даних і оформлення замовлень. Проведено тестування розробленої системи для перевірки її працездатності.

У третьому розділі кваліфікаційної роботи описано працездатність користувача при роботі з комп'ютерною технікою та загальні вимоги охорони праці для користувачів ПК, а також рекомендації щодо безпечної організації робочого місця.

ANNOTATION

Development of a web platform for online ordering and delivery of food products // Qualification work of the educational level "Bachelor" // Maksymchuk Anton // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group SN-42 // Ternopil, 2026 // P. 60, fig. – 15 , add. – 3, bibliography – 46.

Keywords: site, software complex, goods, react, node, express.js.

The qualification work is devoted to the development of a web platform for online ordering and delivery of food products.

The first section of the qualification work describes the statement of the task and the formation of requirements for the web platform. The subject area of the study is analyzed, the main users of the system and usage options are determined. Existing analogues of food delivery systems are considered, and the choice of technologies for the implementation of the project is justified.

In the second section of the qualification work, the general structure of the web platform is formed, a database is designed and created. The user interface and administrative panel have been developed, the main functions of data processing and order processing have been implemented. The developed system has been tested to verify its operability.

The third section of the qualification work describes the user's operability when working with computer equipment and general occupational safety requirements for PC users, as well as recommendations for the safe organization of the workplace.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSS (англ. Cascading Style Sheets) – спеціальна мова, яка використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

HTML (англ. HyperText Markup Language) – стандартизована мова розмітки документів у Всесвітній павутині.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОГО КОМПЛЕКСУ	10
1.1 Аналіз предметної області.....	10
1.2 Формування вимог до програмного комплексу	11
1.3 Пошук актантів та варіантів використання	12
1.4 Вибір середовища розробки	17
1.5 Обґрунтування використовуваних технологій.....	19
1.6 Висновок до першого розділу	21
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА МЕТОДИ РОЗРОБКИ ВЕБПЛАТФОРМИ	23
2.1 Формування структури програмного комплексу	23
2.2 Проектування архітектури програмного комплексу	24
2.3 Проектування моделі взаємодії компонентів системи	26
2.4 Проектування та створення БД для розроблюваного комплексу.....	29
2.5 Розробка алгоритму функціонування вебплатформи	31
2.6 Висновок до другого розділу	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ.....	34
3.1 Проектування інтерфейсу користувача.....	34
3.2 Розробка програмного комплексу	35
3.3 Тестування розробленого програмного комплексу	42
3.4 Перспективи модернізації.....	45
3.5 Висновок до другого розділу	46
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ ...	47
4.1 Заходи щодо автоматизації виробничих процесів, які сприяють покращенню умов праці	47
4.2 Естетичне оформлення робочого місця опеатора ПК, станка, установки.....	49
4.3 Висновок до четвертого розділу	52
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ	56
ДОДАТКИ	

ВСТУП

Актуальність теми. Внаслідок стрімкого розвитку інформаційних технологій та поширення онлайн-сервісів, все більшої популярності набувають вебплатформи для онлайн-замовлення та доставки харчових продуктів. Такі системи дозволяють автоматизувати процес оформлення замовлень, спростити взаємодію між користувачами та сервісом доставки, а також значно підвищити швидкість обробки інформації.

Сучасні користувачі все частіше віддають перевагу онлайн-замовленню їжі через зручність використання, економію часу та можливість швидкого перегляду доступних товарів. Використання вебплатформи дозволяє клієнтам переглядати каталог продукції, здійснювати пошук необхідних товарів, додавати їх до кошика та оформлювати замовлення у декілька кліків.

Для адміністратора системи такі вебплатформи забезпечують зручне управління товарами, категоріями та замовленнями, що значно спрощує процес адміністрування сервісу доставки. Тому розробка вебплатформи для онлайн-замовлення та доставки харчових продуктів є актуальною задачею.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка вебплатформи для онлайн-замовлення та доставки харчових продуктів. Для досягнення поставленої мети необхідно виконати ряд завдань, зокрема:

- проаналізувати предметну область;
- визначити функціональні можливості системи;
- спроектувати структуру програмного комплексу;
- спроектувати та розробити базу даних;
- спроектувати інтерфейс користувача;
- розробити адміністративну панель;
- реалізувати функціонал роботи з товарами та замовленнями;
- провести тестування програмного комплексу.

Практичне значення одержаних результатів. Розроблений у процесі виконання кваліфікаційної роботи програмний комплекс для онлайн-замовлення та доставки харчових продуктів дозволить автоматизувати процес оформлення та обробки замовлень. Вебплатформа забезпечує користувачам можливість швидкого перегляду товарів, оформлення замовлень та отримання актуальної інформації про доступну продукцію.

Для адміністратора системи реалізовано можливість управління товарами, категоріями та замовленнями через адміністративну панель. Це значно спрощує процес адміністрування вебплатформи та дозволяє ефективно контролювати роботу сервісу доставки.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОГО КОМПЛЕКСУ

1.1 Аналіз предметної області

Сучасний розвиток інформаційних технологій та електронної комерції суттєво вплинув на сферу продажу товарів і надання послуг. Одним із найбільш популярних напрямів стала організація онлайн замовлення та доставки харчових продуктів через вебплатформи та мобільні застосунки. Такий підхід дозволяє користувачам швидко оформлювати замовлення, переглядати асортимент продукції, оплачувати покупки онлайн та отримувати доставку у зручне місце й час.

Предметною областю даного дослідження є процес автоматизації замовлення та доставки харчових продуктів із використанням вебтехнологій. Основною метою таких систем є забезпечення ефективної взаємодії між клієнтом, адміністратором і службою доставки. Вебплатформа повинна забезпечувати зручний інтерфейс, високу швидкість роботи, безпечне зберігання даних та можливість обробки великої кількості замовлень.

На сьогодні існує значна кількість сервісів онлайн доставки продуктів та готової їжі, серед яких Glovo, Bolt Food та Uber Eats. Дані системи надають користувачам можливість швидкого пошуку товарів, оформлення замовлень та відстеження доставки в режимі реального часу. Проте багато існуючих платформ мають недоліки, серед яких складність адміністрування, перевантажений інтерфейс, високе навантаження на систему або недостатня адаптація під потреби окремих закладів чи магазинів.

Таким чином, аналіз предметної області показує, що розробка вебплатформи для онлайн замовлення та доставки харчових продуктів є актуальним та перспективним напрямом. Створення такої системи дозволить автоматизувати процеси продажу, покращити якість обслуговування клієнтів та забезпечити ефективне управління замовленнями.

1.2 Формування вимог до програмного комплексу

Під час розробки вебплатформи для онлайн замовлення та доставки харчових продуктів важливим етапом є формування вимог до програмного комплексу. Саме вимоги визначають основні функції системи, її структуру, особливості взаємодії користувачів із платформою, а також забезпечують ефективність та зручність використання програмного продукту.

Основною метою програмного комплексу є автоматизація процесу замовлення харчових продуктів через мережу Інтернет із можливістю швидкого оформлення покупки, управління товарами та обробки замовлень адміністраторами системи. Розроблювана вебплатформа повинна забезпечувати стабільну роботу, зручний інтерфейс та можливість масштабування у майбутньому.

До функціональних вимог програмного комплексу належать:

- реєстрація та авторизація користувачів;
- перегляд каталогу товарів;
- пошук продукції за назвою або категорією;
- фільтрація товарів;
- додавання товарів до кошика;
- оформлення онлайн замовлення;
- зміна кількості товарів у кошику;
- збереження інформації про замовлення;
- адміністрування товарів і категорій;
- редагування та видалення товарів;
- управління статусами замовлень;
- можливість перегляду історії замовлень.

Система повинна підтримувати декілька ролей користувачів. Основними ролями є:

- користувач;
- адміністратор.

Користувач отримує доступ до каталогу продукції, функцій оформлення замовлення та перегляду інформації про свої покупки. Адміністратор має розширені можливості, зокрема додавання нових товарів, редагування інформації, перегляд усіх замовлень та управління роботою системи.

Важливими є також нефункціональні вимоги до програмного комплексу. До них належать:

- висока швидкість роботи вебплатформи;
- адаптивність інтерфейсу для різних пристроїв;
- надійність зберігання даних;
- безпека персональної інформації користувачів;
- простота користування системою;
- зручний та сучасний дизайн;
- можливість подальшого розширення функціоналу.

Для реалізації програмного комплексу доцільно використовувати сучасні вебтехнології. Клієнтська частина системи може бути реалізована за допомогою React, що забезпечує швидке оновлення інтерфейсу та зручну взаємодію користувача із системою. Серверна частина може бути реалізована на основі Node.js та Express.js, які дозволяють створити ефективний API для обробки запитів. Для зберігання інформації доцільно використовувати MongoDB, оскільки дана система управління базами даних добре підходить для вебзастосунків із великою кількістю динамічних даних.

Таким чином, сформовані вимоги до програмного комплексу визначають основні функції та характеристики вебплатформи для онлайн замовлення та доставки харчових продуктів. Їх врахування під час розробки дозволить створити ефективний, сучасний та зручний програмний продукт.

1.3 Пошук актантів та варіантів використання

Для пошуку усіх можливих варіантів використання вебплатформи для онлайн замовлення та доставки харчових продуктів доцільно побудувати

діаграму варіантів використання. Дана діаграма дозволяє визначити основних користувачів системи, їх функції та взаємодію із програмним комплексом.

У ролі акторів було обрано:

- адміністратора системи;
- зареєстрованого користувача;
- незареєстрованого користувача;
- кур'єра служби доставки.

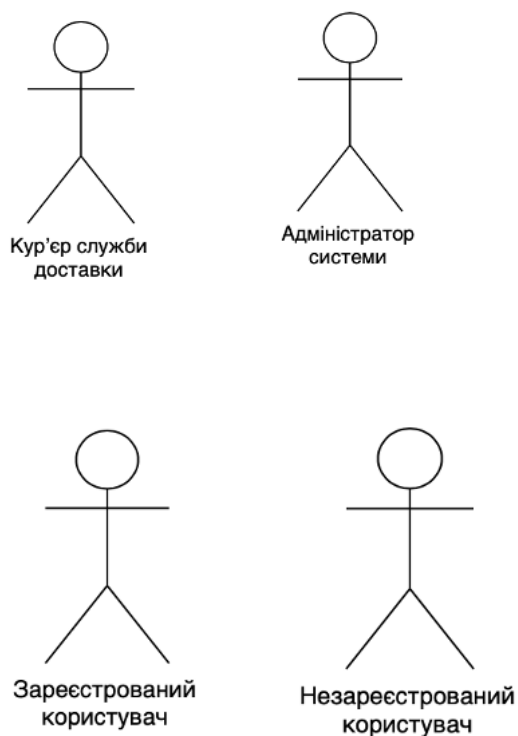


Рисунок 1.1 — Актори діаграми варіантів використання

Для актора «Адміністратор системи» було розроблено такі варіанти використання:

- авторизація в системі;
- додавання нових товарів;
- редагування інформації про товари;
- видалення товарів;
- управління категоріями продукції;

- перегляд списку користувачів;
- перегляд та обробка замовлень;
- зміна статусу доставки;
- управління роботою вебплатформи.

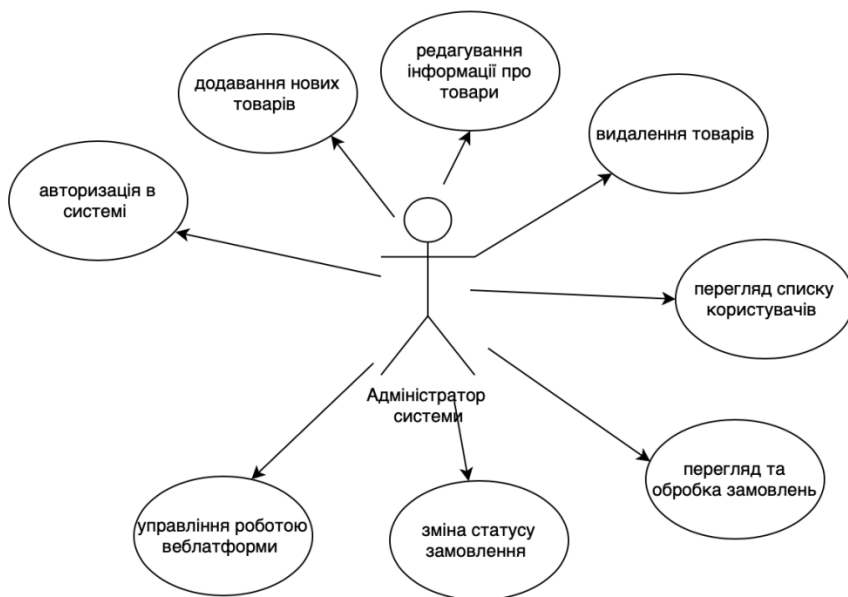


Рисунок 1.2 — Варіанти використання для адміністратора системи

Адміністратор має найбільший рівень доступу до програмного комплексу. Його основним завданням є забезпечення стабільної роботи вебплатформи, контроль замовлень та редагування інформації про продукцію. За допомогою адміністративної панелі адміністратор може оперативно змінювати дані системи та здійснювати контроль за процесом доставки.

Для актора «Зареєстрований користувач» було визначено такі варіанти використання:

- реєстрація та авторизація;
- перегляд каталогу товарів;
- пошук продукції;
- фільтрація товарів за категоріями;

- додавання товарів до кошика;
- оформлення замовлення;
- перегляд історії замовлень;
- редагування особистих даних;
- перегляд статусу доставки.

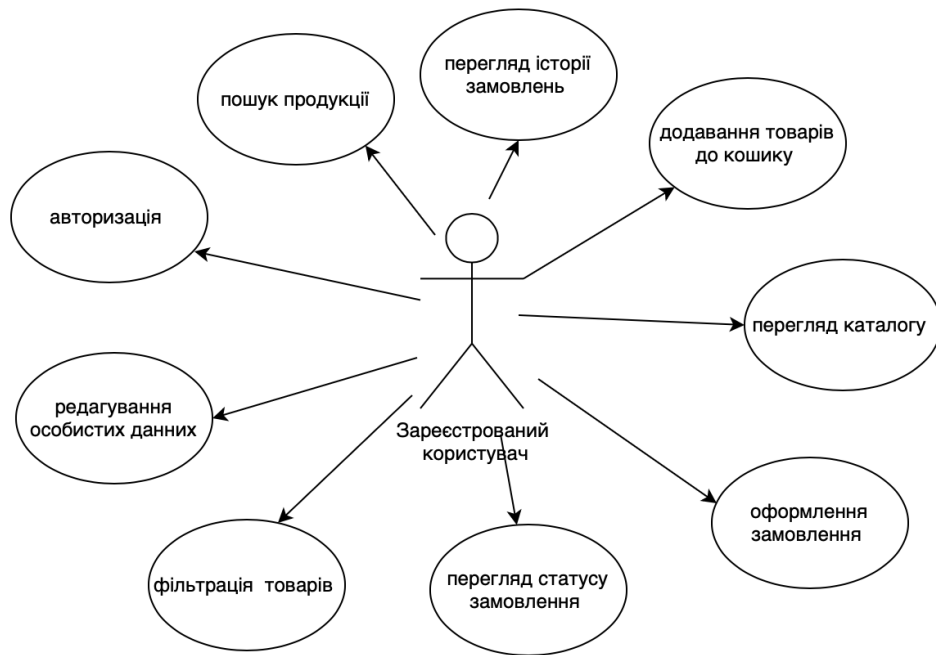


Рисунок 1.3 — Варіанти використання для зареєстрованого користувача

Основним призначенням користувача є взаємодія із системою для здійснення онлайн покупок. Користувач має змогу переглядати доступні товари, додавати продукцію до кошика та оформлювати доставку через вебплатформу. Також користувач може отримувати інформацію про статус свого замовлення.

Для актора «Незареєстрований користувач» було передбачено такі варіанти використання:

- перегляд головної сторінки;
- перегляд каталогу товарів;
- пошук продукції;
- реєстрація в системі.



Рисунок 1.4 — Варіанти використання для незареєстрованого користувача

Незареєстрований користувач має обмежений функціонал системи. Він може лише переглядати доступну інформацію про товари та виконати реєстрацію для отримання повного доступу до функцій вебплатформи.

Для актора «Кур'єр служби доставки» було визначено такі варіанти використання:

- перегляд нових замовлень;
- отримання інформації про доставку;
- зміна статусу замовлення;
- підтвердження виконання доставки.

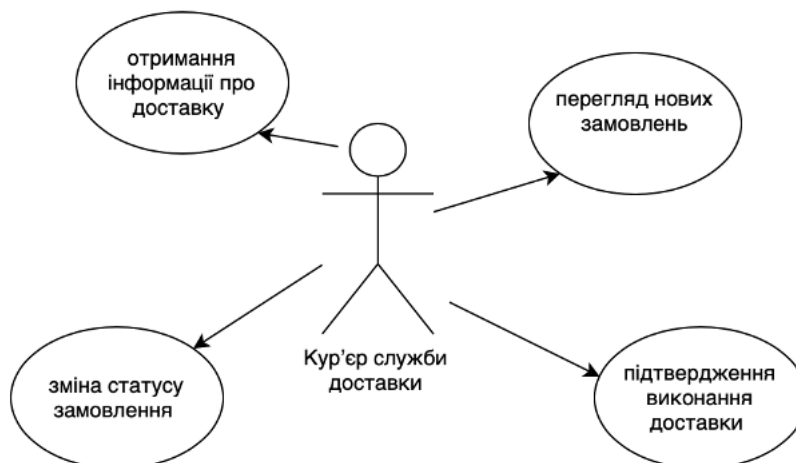


Рисунок 1.5 — Варіанти використання для кур'єра служби доставки

Кур'єр відповідає за доставку продукції користувачам та оновлення інформації про стан виконання замовлення. Завдяки цьому користувачі можуть відстежувати процес доставки у режимі реального часу.

Отже, побудова діаграми варіантів використання дозволяє визначити основні функції програмного комплексу та взаємодію між акторами системи. Це забезпечує правильне проєктування вебплатформи та формування її функціональної структури [1-2].

Кожен актор має певний ряд дій, які він може виконати. Найбільше прав має головний адміністратор, а найменше – незареєстрований користувач.

1.4 Вибір середовища розробки

Одним із важливих етапів створення програмного комплексу є вибір середовища розробки та допоміжних програмних засобів. Від правильності вибору залежить швидкість розробки, зручність написання коду, тестування системи та подальша підтримка програмного продукту.

Для розробки вебплатформи онлайн замовлення та доставки харчових продуктів було обрано сучасні інструменти та середовища розробки, які

забезпечують ефективну роботу із клієнтською та серверною частинами системи.

Основним середовищем розробки було обрано Visual Studio Code. Даний редактор коду є одним із найпопулярніших серед веброзробників завдяки високій швидкості роботи, підтримці великої кількості мов програмування та наявності широкого набору розширень. Середовище підтримує інтеграцію із системами контролю версій, автоматичне форматування коду, підсвічування синтаксису та налагодження програмного забезпечення.

Основними перевагами використання Visual Studio Code є:

- простий та зручний інтерфейс;
- підтримка сучасних вебтехнологій;
- велика кількість доступних плагінів;
- інтеграція з Git;
- підтримка роботи з TypeScript та JavaScript;
- вбудований термінал;
- можливість запуску та тестування проєкту безпосередньо в

редакторі.

Для розробки клієнтської частини вебплатформи використовувалась бібліотека React, яка дозволяє створювати сучасні інтерактивні користувацькі інтерфейси. React забезпечує компонентний підхід до розробки, що значно спрощує підтримку та масштабування проєкту.

Серверна частина програмного комплексу реалізовувалась за допомогою Node.js та фреймворку Express.js. Використання даних технологій дозволяє створювати швидкі та ефективні серверні застосунки для обробки HTTP-запитів та взаємодії з базою даних.

Для зберігання інформації про користувачів, товари та замовлення було обрано MongoDB. Дана система керування базами даних забезпечує гнучке зберігання інформації та добре підходить для вебзастосунків із великою кількістю динамічних даних[3].

Під час тестування вебплатформи використовувалися сучасні веббраузери, зокрема Google Chrome та Mozilla Firefox. Це дозволило перевірити правильність роботи користувацького інтерфейсу та адаптивність вебзастосунку на різних пристроях.

Для контролю версій програмного коду використовувалась система Git. Вона забезпечує збереження історії змін проєкту, спрощує командну роботу та дозволяє швидко відновлювати попередні версії коду.

Таким чином, обрані середовища розробки та програмні засоби забезпечують ефективну реалізацію вебплатформи для онлайн замовлення та доставки харчових продуктів, а також дозволяють створити сучасний, масштабований та зручний програмний продукт.

1.5 Обґрунтування використовуваних технологій

Під час розробки вебплатформи для онлайн замовлення та доставки харчових продуктів важливим етапом є вибір та обґрунтування використовуваних технологій. Від правильності вибору залежить продуктивність системи, швидкість розробки, стабільність роботи та можливість подальшого масштабування програмного комплексу.

Для створення клієнтської частини вебплатформи було обрано бібліотеку React. Дана технологія є однією з найпопулярніших у сфері веброзробки та широко використовується для створення сучасних односторінкових застосунків. Основною перевагою React є компонентний підхід, який дозволяє розділити інтерфейс на окремі незалежні компоненти. Це значно спрощує підтримку програмного коду та повторне використання елементів інтерфейсу [4].

Використання React забезпечує:

- швидке оновлення інтерфейсу без перезавантаження сторінки;
- високу продуктивність застосунку;
- зручну організацію структури проєкту;
- можливість створення адаптивного дизайну;
- спрощення масштабування програмного комплексу.

Для реалізації серверної частини системи було обрано Node.js та фреймворк Express.js. Node.js дозволяє виконувати JavaScript-код на стороні сервера та забезпечує високу швидкість обробки запитів завдяки асинхронній моделі роботи. Це особливо важливо для вебплатформи онлайн замовлення, де система повинна одночасно обробляти велику кількість користувацьких запитів.

Фреймворк Express.js було обрано через:

- простоту створення REST API;
- швидку обробку HTTP-запитів;
- зручну організацію маршрутів;
- підтримку middleware;
- можливість інтеграції з різними базами даних.

Для зберігання інформації про користувачів, товари, категорії та замовлення використовується MongoDB. Дана система управління базами даних є документоорієнтованою та дозволяє гнучко працювати зі структурами даних. MongoDB добре підходить для вебзастосунків, у яких структура інформації може змінюватися або доповнюватися.

Основними перевагами MongoDB є:

- висока швидкість роботи;
- гнучкість структури даних;
- простота масштабування;
- зручна інтеграція з Node.js;
- підтримка роботи з JSON-подібними документами.

Для стилізації інтерфейсу вебплатформи використовується Tailwind CSS. Даний CSS-фреймворк дозволяє швидко створювати сучасний адаптивний дизайн без написання великої кількості власних CSS-стилів. Tailwind CSS забезпечує гнучкість у створенні інтерфейсу та значно прискорює процес розробки[5].

Для взаємодії клієнтської та серверної частини системи використовуються HTTP-запити та формат обміну даними JSON. Це забезпечує швидкий обмін

інформацією між компонентами програмного комплексу та дозволяє ефективно реалізувати функції авторизації, оформлення замовлень та управління товарами.

Під час розробки також використовувалась система контролю версій Git, яка дозволяє зберігати історію змін програмного коду, спрощує командну роботу та забезпечує можливість відновлення попередніх версій проєкту.

Отже, обрані технології є сучасними, ефективними та добре підходять для створення вебплатформи онлайн замовлення та доставки харчових продуктів. Їх використання забезпечує високу продуктивність системи, зручність користування та можливість подальшого розвитку програмного комплексу.

1.6 Висновок до першого розділу

У першому розділі було проведено аналіз предметної області вебплатформ для онлайн замовлення та доставки харчових продуктів. Визначено основні особливості функціонування подібних систем, їх переваги та актуальність у сучасних умовах розвитку електронної комерції. Аналіз існуючих рішень показав, що вебплатформи доставки продуктів користуються значним попитом серед користувачів завдяки зручності, швидкості оформлення замовлень та можливості дистанційного придбання товарів.

У ході дослідження було сформовано основні функціональні та нефункціональні вимоги до програмного комплексу. Визначено, що система повинна забезпечувати можливість реєстрації користувачів, перегляду каталогу продукції, пошуку товарів, оформлення замовлень та адміністрування вебплатформи. Особливу увагу приділено питанням безпеки даних, адаптивності інтерфейсу та зручності користування системою.

Також у першому розділі було визначено основних актантів системи та варіанти використання програмного комплексу. Це дозволило описати взаємодію між користувачами, адміністраторами та службою доставки, а також сформулювати загальну логіку роботи вебплатформи. Побудова діаграм варіантів використання дала змогу наочно відобразити функціональні можливості системи та структуру взаємодії між її компонентами.

Крім того, було виконано вибір середовища розробки та обґрунтовано використання сучасних технологій веброзробки. Для створення програмного комплексу обрано React, Node.js, Express.js та MongoDB, які забезпечують високу продуктивність, масштабованість та ефективність роботи вебплатформи.

Отже, результати першого розділу створюють необхідну основу для подальшого проєктування, реалізації та тестування вебплатформи онлайн замовлення та доставки харчових продуктів.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА МЕТОДИ РОЗРОБКИ ВЕБПЛАТФОРМИ

2.1 Формування структури програмного комплексу

Формування структури програмного комплексу є важливим етапом проєктування вебплатформи для онлайн замовлення та доставки харчових продуктів. На даному етапі визначається загальна архітектура системи, взаємодія між її основними компонентами, а також розподіл функціональних модулів, що забезпечують коректну роботу програмного продукту.

Розроблювана вебплатформа побудована за клієнт-серверною архітектурою, що є типовим підходом для сучасних вебзастосунків. Така архітектура передбачає поділ системи на дві основні частини: клієнтську (frontend) та серверну (backend), які взаємодіють між собою через HTTP-запити з використанням формату JSON.

Клієнтська частина відповідає за відображення інтерфейсу користувача та забезпечення взаємодії з системою. Вона реалізована із використанням React, що дозволяє створювати динамічні та інтерактивні компоненти інтерфейсу. До основних модулів клієнтської частини належать:

- модуль авторизації та реєстрації користувачів;
- модуль каталогу товарів;
- модуль кошика;
- модуль оформлення замовлення;
- модуль перегляду історії замовлень;
- адміністративна панель.

Серверна частина програмного комплексу реалізована на базі Node.js із використанням фреймворку Express.js. Вона відповідає за обробку запитів, бізнес-логіку системи та взаємодію з базою даних. Основними модулями серверної частини є модуль адміністративного управління.

Для зберігання даних використовується MongoDB, яка забезпечує гнучку структуру зберігання інформації та високу продуктивність при роботі з великими

обсягами даних. У базі даних зберігаються колекції користувачів, товарів, замовлень та категорій.

Структуру програмного комплексу можна умовно поділити на такі рівні:

- рівень представлення (frontend);
- рівень бізнес-логіки (backend);
- рівень збереження даних (database).

Кожен із цих рівнів виконує окрему функцію та взаємодіє з іншими через чітко визначені інтерфейси. Такий підхід забезпечує модульність системи, спрощує її підтримку та дозволяє у майбутньому розширювати функціонал без значних змін у вже реалізованих компонентах.

Отже, сформована структура програмного комплексу забезпечує ефективну організацію взаємодії між основними модулями вебплатформи та створює основу для подальшої реалізації функціональних можливостей системи.

2.2 Проекування архітектури програмного комплексу

Проекування архітектури програмного комплексу є одним із найважливіших етапів створення вебплатформи для онлайн-замовлення та доставки харчових продуктів, оскільки саме на цьому етапі визначається структура системи, взаємозв'язок її компонентів та принципи обробки інформації. Грамотно побудована архітектура забезпечує високу продуктивність, надійність, масштабованість та можливість подальшого розвитку програмного комплексу.

Розроблювана вебплатформа побудована за клієнт-серверною архітектурою, що передбачає розподіл функцій між клієнтською та серверною частинами. Такий підхід дозволяє забезпечити швидку обробку запитів користувачів, централізоване зберігання даних та ефективне управління бізнес-процесами системи.

Клієнтська частина реалізована у вигляді вебзастосунку, який надає користувачу можливість переглядати каталог товарів, здійснювати пошук

продукції, додавати товари до кошика, оформлювати замовлення та відстежувати їх виконання. Інтерфейс побудований із використанням сучасних вебтехнологій та адаптований для роботи на персональних комп'ютерах, планшетах і мобільних пристроях.

Серверна частина реалізована на платформі Node.js із використанням фреймворку Express.js. Вона відповідає за обробку HTTP-запитів, виконання бізнес-логіки, автентифікацію користувачів, взаємодію з базою даних та формування відповідей для клієнтської частини. Завдяки асинхронній моделі роботи Node.js забезпечується висока швидкодія навіть при одночасному обслуговуванні великої кількості користувачів.

Для зберігання інформації використовується база даних MongoDB, у якій містяться відомості про користувачів, категорії товарів, продукти, кошики, замовлення та історію покупок. Документоорієнтована структура бази даних дозволяє швидко виконувати операції пошуку та оновлення інформації, а також легко масштабувати систему при збільшенні кількості користувачів.

Обмін інформацією між клієнтською та серверною частинами здійснюється через REST API із використанням формату JSON. Кожен запит користувача передається на сервер, де виконується його обробка та формування відповіді, після чого результат повертається клієнтській частині для відображення на екрані.

Архітектура програмного комплексу побудована за модульним принципом. До її складу входять модулі управління користувачами, товарами, кошиком, замовленнями, оплатою та доставкою. Така організація програмного забезпечення спрощує супровід системи, забезпечує незалежність окремих компонентів та дозволяє без значних змін інтегрувати нові функціональні можливості.

Архітектура вебплатформи наведена на рисунку 2.1, де схематично відображено взаємодію користувача, клієнтської частини, серверної частини, функціональних модулів та бази даних. Представлена схема демонструє основні потоки передачі інформації та логіку роботи програмного комплексу.

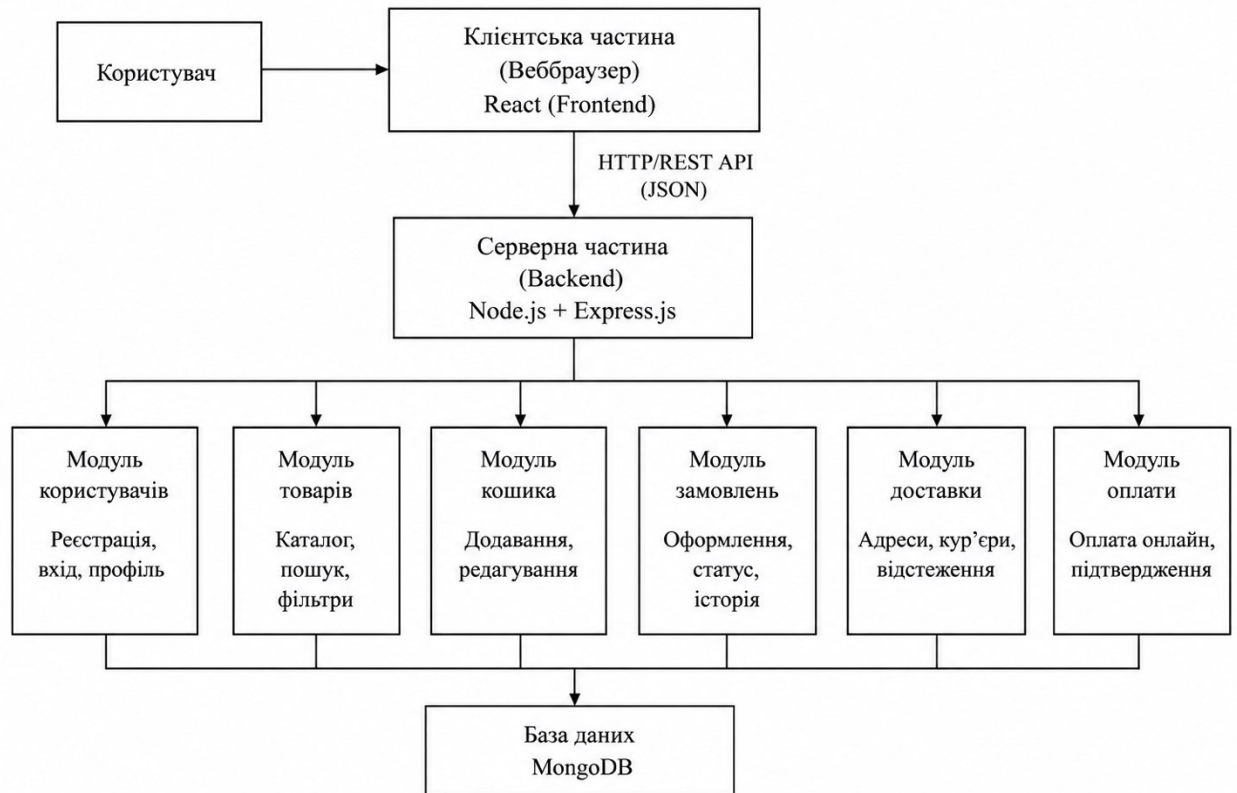


Рисунок 2.1- Архітектура вебплатформи

Отже, запропонована архітектура забезпечує ефективну взаємодію всіх компонентів системи, високу продуктивність, безпечне зберігання інформації та створює основу для подальшого розвитку вебплатформи для онлайн-замовлення і доставки харчових продуктів.

2.3 Проектування моделі взаємодії компонентів системи

Проектування моделі взаємодії компонентів системи є важливим етапом розробки вебплатформи для онлайн-замовлення та доставки харчових продуктів. Саме на цьому етапі визначається порядок обміну інформацією між окремими складовими програмного комплексу, що забезпечує швидке виконання операцій, надійність роботи системи та коректну взаємодію між користувачем, сервером і базою даних.

Модель взаємодії побудована за принципом клієнт-серверної архітектури, у якій усі запити користувача надходять із клієнтської частини до серверної, де

виконується їх обробка та формування відповіді. Такий підхід забезпечує централізоване керування бізнес-логікою та дозволяє ефективно контролювати всі процеси роботи вебплатформи.

Після відкриття вебплатформи користувач взаємодіє з інтерфейсом через веббраузер. Під час перегляду каталогу товарів, авторизації, додавання товарів до кошика або оформлення замовлення клієнтська частина формує HTTP-запит та передає його до серверної частини за допомогою REST API.

Серверний застосунок приймає запит, аналізує його, перевіряє права доступу користувача та визначає необхідний модуль для виконання відповідної операції. Залежно від типу запиту система може звертатися до модулів керування товарами, користувачами, кошиком, замовленнями або оплатою.

У разі необхідності отримання або оновлення інформації сервер взаємодіє з базою даних, де зберігаються відомості про користувачів, товари, категорії, оформлені замовлення та історію покупок. Після виконання операції база даних повертає результат серверній частині, яка формує відповідь у форматі JSON.

Отримана інформація передається клієнтській частині, де вона автоматично відображається користувачу без перезавантаження сторінки. Такий механізм забезпечує швидку роботу вебплатформи, покращує користувацький досвід та дозволяє оперативно виконувати всі необхідні операції.

Для забезпечення безпечної взаємодії між компонентами системи використовується механізм автентифікації та авторизації користувачів. Після успішного входу до системи сервер формує токен доступу, який використовується для підтвердження особи користувача під час виконання наступних запитів.

Завдяки модульній організації системи окремі компоненти можуть працювати незалежно один від одного, що значно спрощує супровід програмного забезпечення та дозволяє розширювати його функціональність без внесення суттєвих змін до вже реалізованих модулів.

Модель взаємодії компонентів вебплатформи наведено на **рисунку 2.2**, де відображено послідовність обміну інформацією між користувачем, клієнтською частиною, сервером застосунку та базою даних. Представлена схема демонструє

логіку функціонування програмного комплексу та взаємозв'язок між його основними складовими.



Рисунок 2.2 - Модель взаємодії вебплатформи

Отже, спроектована модель взаємодії компонентів забезпечує ефективну організацію роботи вебплатформи, швидкий обмін інформацією між її елементами та створює надійну основу для стабільного функціонування системи онлайн-замовлення і доставки харчових продуктів.

2.4 Проєктування та створення БД для розроблюваного комплексу

Проєктування бази даних є одним із ключових етапів розробки вебплатформи для онлайн замовлення та доставки харчових продуктів, оскільки саме вона забезпечує зберігання, обробку та цілісність інформації про користувачів, товари та замовлення. Від правильної структури бази даних залежить продуктивність системи, швидкість виконання запитів та зручність подальшого масштабування програмного комплексу.

Для реалізації бази даних у розроблюваній вебплатформі було обрано MongoDB. Дана система керування базами даних є документоорієнтованою та дозволяє зберігати інформацію у форматі JSON-подібних документів. Такий підхід є особливо зручним для вебзастосунків, де структура даних може змінюватися або доповнюватися в процесі розвитку проєкту[6].

Основними перевагами використання MongoDB є:

- гнучка схема даних (відсутність жорсткої структури таблиць);
- висока швидкість виконання запитів;
- простота масштабування системи;
- зручна інтеграція з серверною частиною на базі Node.js;
- підтримка роботи з великими обсягами інформації.

На основі аналізу предметної області та визначених вимог до програмного комплексу була спроєктована логічна структура бази даних вебплатформи для онлайн замовлення та доставки харчових продуктів. Дана структура відображає основні сутності системи, їх атрибути та взаємозв'язки між ними, що забезпечує коректне зберігання та обробку інформації.

База даних включає такі основні сутності: користувачі (Users), товари (Products), категорії (Categories), замовлення (Orders), елементи замовлення (OrderItems), кошик (Cart) та платежі (Payments). Кожна сутність містить набір полів, які описують її властивості, а також забезпечують взаємодію між різними частинами системи.

Взаємозв'язки між сутностями побудовані таким чином, що один користувач може створювати декілька замовлень, кожне замовлення містить один або декілька товарів, а кожен товар належить до певної категорії. Крім того, для кожного замовлення може бути створений запис про оплату, що дозволяє відслідковувати статус фінансових операцій у системі.

Графічне представлення структури бази даних наведено на рисунку 2.1.

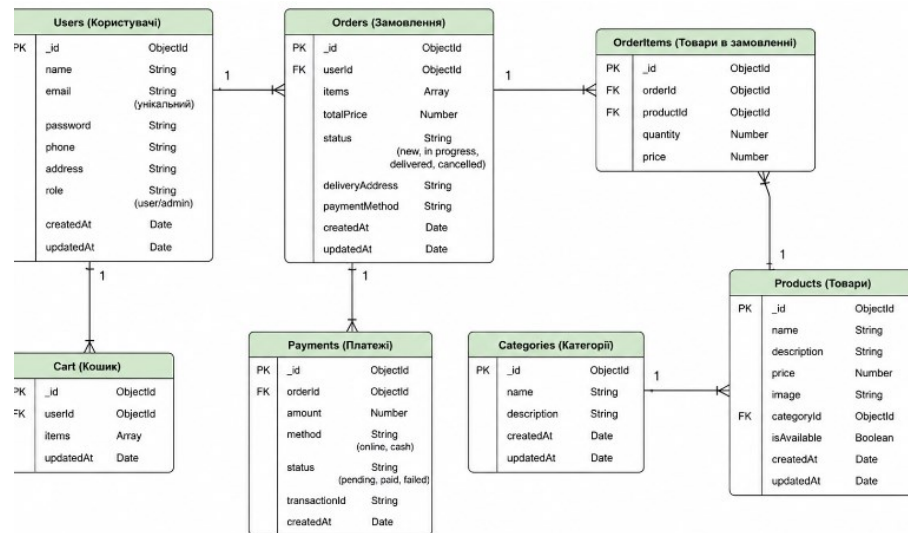


Рисунок 2.3 — Діаграма бази даних вебплатформи онлайн замовлення та доставки харчових продуктів

На рисунку 2.3 зображено структуру бази даних вебплатформи для онлайн замовлення та доставки харчових продуктів. Вона складається з основних сутностей: `Users`, `Products`, `Categories`, `Orders` та `OrderItems`. Користувачі пов'язані із замовленнями через зв'язок один-до-багатьох. Замовлення містять перелік товарів, який реалізовано через проміжну таблицю `OrderItems`. Товари належать до певних категорій, що дозволяє систематизувати їх у базі даних. Така структура забезпечує коректне зберігання інформації та зв'язки між основними елементами системи.

2.5 Розробка алгоритму функціонування вебплатформи

Розробка алгоритму функціонування вебплатформи є одним із ключових етапів проєктування програмного комплексу, оскільки саме алгоритм визначає послідовність виконання всіх операцій, взаємодію між користувачем і системою та порядок обробки інформації. Чітко побудований алгоритм забезпечує стабільну роботу вебплатформи, швидке виконання запитів та ефективно обслуговування користувачів.

Робота програмного комплексу починається з відкриття вебплатформи користувачем. Після завантаження головної сторінки система надає можливість перегляду каталогу товарів, пошуку продукції, авторизації або реєстрації нового облікового запису.

У разі авторизації користувач вводить свої облікові дані, які передаються на сервер для перевірки. Після успішної автентифікації користувач отримує доступ до персонального кабінету та можливість оформлення замовлень.

Наступним етапом є вибір товарів із каталогу. Користувач може переглядати категорії продукції, застосовувати фільтри, здійснювати пошук та додавати необхідні товари до кошика. Усі дії автоматично обробляються серверною частиною та синхронізуються з базою даних.

Після формування кошика користувач переходить до оформлення замовлення, де заповнює контактні дані, обирає адресу доставки та спосіб оплати. Отримана інформація перевіряється сервером на коректність, після чого формується новий запис про замовлення у базі даних.

Після успішного оформлення система змінює статус замовлення та відображає користувачу повідомлення про його прийняття. Надалі адміністратор має можливість переглядати нові замовлення, змінювати їх статус та контролювати процес доставки.

Алгоритм функціонування вебплатформи побудований циклічно. Після завершення кожної операції користувач може продовжити роботу із системою, оформити нове замовлення, переглянути історію покупок або завершити сеанс роботи.

Завдяки модульному принципу побудови алгоритму окремі функції програмного комплексу працюють незалежно одна від одної, що спрощує подальшу модернізацію системи та інтеграцію нових сервісів без зміни основної логіки роботи вебплатформи.

Отже, розроблений алгоритм функціонування забезпечує ефективну організацію роботи вебплатформи, швидку обробку інформації, коректне виконання всіх бізнес-процесів та створює основу для стабільної роботи системи онлайн-замовлення і доставки харчових продуктів.

2.6 Висновок до другого розділу

У другому розділі кваліфікаційної роботи було розглянуто основні етапи проєктування вебплатформи для онлайн-замовлення та доставки харчових продуктів. На основі проведеного аналізу сформовано структуру програмного комплексу, визначено його основні складові та принципи взаємодії між ними.

У процесі виконання роботи було спроектовано архітектуру програмного комплексу, яка базується на клієнт-серверній моделі та забезпечує ефективний розподіл функцій між клієнтською частиною, сервером застосунку та базою даних. Запропонована архітектура дозволяє забезпечити високу продуктивність системи, її масштабованість та можливість подальшого розвитку.

Також було розроблено модель взаємодії компонентів вебплатформи, яка визначає порядок обміну інформацією між користувачем, клієнтською частиною, сервером та базою даних. Реалізація такої моделі забезпечує швидке виконання запитів, централізовану обробку інформації та стабільне функціонування програмного комплексу.

Особливу увагу приділено проєктуванню бази даних, яка забезпечує зберігання інформації про користувачів, товари, кошики та замовлення. Побудована структура бази даних відповідає функціональним вимогам системи та створює основу для ефективної роботи всіх її модулів.

Крім того, було розроблено алгоритм функціонування вебплатформи, який визначає послідовність виконання операцій під час взаємодії користувача із системою — від перегляду товарів до оформлення та підтвердження замовлення. Запропонований алгоритм забезпечує логічну організацію роботи програмного комплексу та підвищує ефективність його використання.

Отже, виконане проєктування дозволило сформувати цілісну модель майбутньої вебплатформи, визначити принципи її функціонування та підготувати необхідну основу для реалізації програмного комплексу й подальшого тестування його роботи.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Проєктування інтерфейсу користувача

Проєктування інтерфейсу користувача є важливим етапом розробки вебплатформи для онлайн замовлення та доставки харчових продуктів, оскільки саме він забезпечує взаємодію користувача із системою. Основною метою інтерфейсу є забезпечення простоти використання, інтуїтивної навігації та швидкого доступу до основних функцій платформи.

Під час розробки інтерфейсу було враховано принципи зручності (usability), мінімалізму та адаптивності. Інтерфейс повинен коректно відображатися на різних пристроях, зокрема на персональних комп'ютерах, планшетах та смартфонах.

Структура користувацького інтерфейсу включає основні сторінки:

- головна сторінка з переліком товарів;
- сторінка категорій;
- сторінка перегляду товару;
- кошик користувача;
- сторінка оформлення замовлення;
- сторінка авторизації та реєстрації.

Головна сторінка забезпечує перегляд доступних товарів та швидкий перехід до деталей обраного продукту. Сторінка товару містить детальний опис, ціну та можливість додавання товару до кошика. Кошик дозволяє користувачу переглядати обрані товари, змінювати їх кількість та переходити до оформлення замовлення [7].

Сторінка оформлення замовлення забезпечує введення контактних даних користувача та підтвердження покупки. Для авторизованих користувачів передбачено можливість перегляду історії замовлень.

Інтерфейс розроблено з використанням сучасних вебтехнологій, що забезпечує швидку взаємодію з серверною частиною та зручність використання

3.2 Розробка програмного комплексу

Розробка програмного комплексу вебплатформи для онлайн замовлення та доставки харчових продуктів включає створення серверної та клієнтської частин системи, а також забезпечення їх взаємодії через API. Основною метою даного етапу є реалізація функціоналу, який забезпечує обробку запитів користувачів, роботу з базою даних та відображення інформації в інтерфейсі.

Серверна частина системи реалізована з використанням платформи Node.js та фреймворку Express.js. Вона відповідає за обробку HTTP-запитів, реалізацію бізнес-логіки, авторизацію користувачів та взаємодію з базою даних MongoDB. Для забезпечення безпеки використовується JWT-аутентифікація, яка дозволяє ідентифікувати користувача та обмежувати доступ до функціоналу системи.

Клієнтська частина розроблена з використанням сучасних JavaScript-бібліотек (React / React Native залежно від реалізації проєкту). Вона забезпечує зручний інтерфейс користувача, відображення товарів, формування кошика та оформлення замовлень. Взаємодія з сервером здійснюється через REST API за допомогою HTTP-запитів [8-9].

У системі реалізовано основні функціональні модулі:

- модуль авторизації та реєстрації користувачів;
- модуль управління товарами та категоріями;
- модуль формування та обробки замовлень;
- модуль кошика користувача;
- модуль адміністрування системи.

Особливу увагу приділено розмежуванню прав доступу. Адміністратор має повний доступ до керування товарами, замовленнями та користувачами, тоді як звичайний користувач може переглядати товари, додавати їх до кошика та оформлювати замовлення.

Таким чином, розроблений програмний комплекс забезпечує повний цикл обробки замовлення — від вибору товару до його підтвердження та збереження в базі даних.

Розробку інтерфейсу користувача вебплатформи для онлайн замовлення та доставки харчових продуктів було розпочато зі створення головного компонента застосунку, який забезпечує навігацію між основними сторінками системи: головною сторінкою, кошиком, формою входу та реєстрації користувача.

Інтерфейс реалізовано з використанням бібліотеки React, що дозволяє будувати компонентну архітектуру та ефективно керувати станом сторінок за допомогою механізму `useState`. Перемикання між сторінками здійснюється через змінну стану `page`, що забезпечує просту логіку навігації без використання додаткових бібліотек маршрутизації [10].

Головна сторінка вебзастосунку містить перелік доступних страв, які відображаються у вигляді карток із зображенням, назвою та ціною. Приклад зовнішнього вигляду головної сторінки наведено на рисунку 3.1.

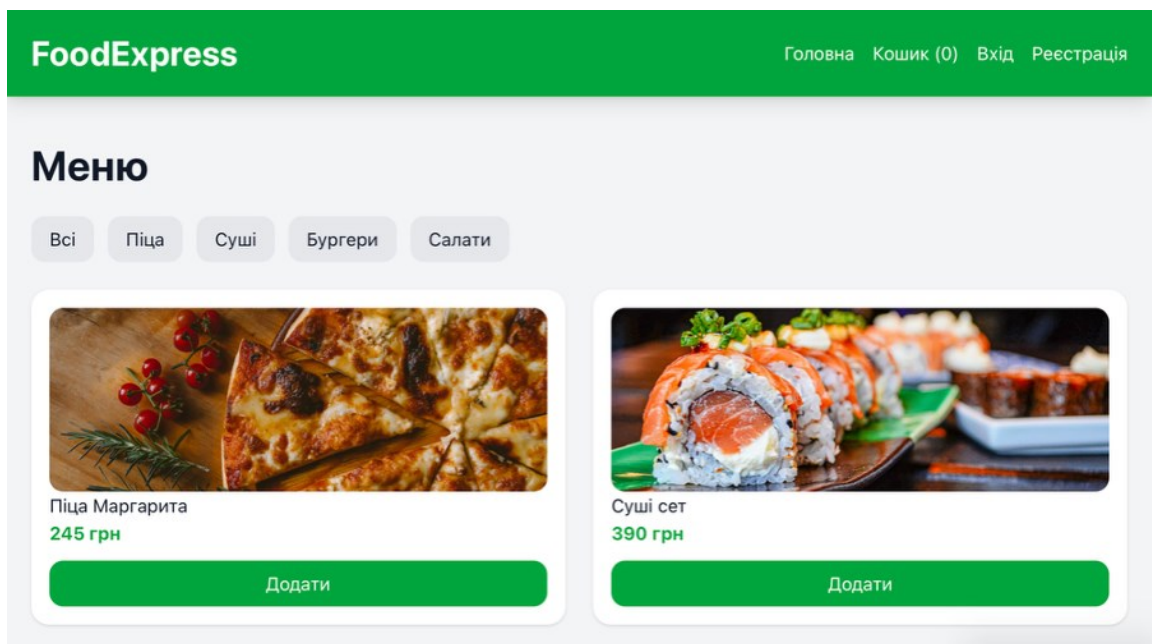


Рисунок 3.1 – Головна сторінка вебплатформи онлайн замовлення їжі

Для зручності користувачів реалізовано систему фільтрації товарів за категоріями (піца, суші, бургери, салати). Фільтрація дозволяє швидко знаходити необхідні позиції меню та зменшує час пошуку товарів. Відображення фільтрів наведено на рисунку 2.2.

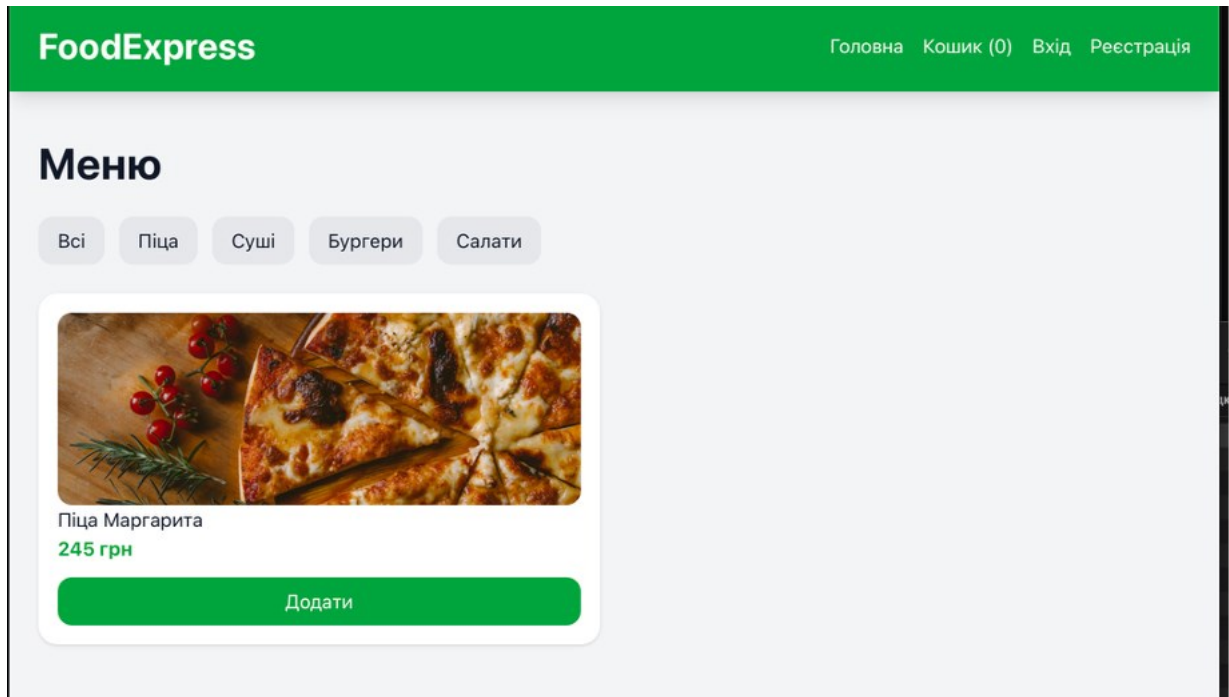


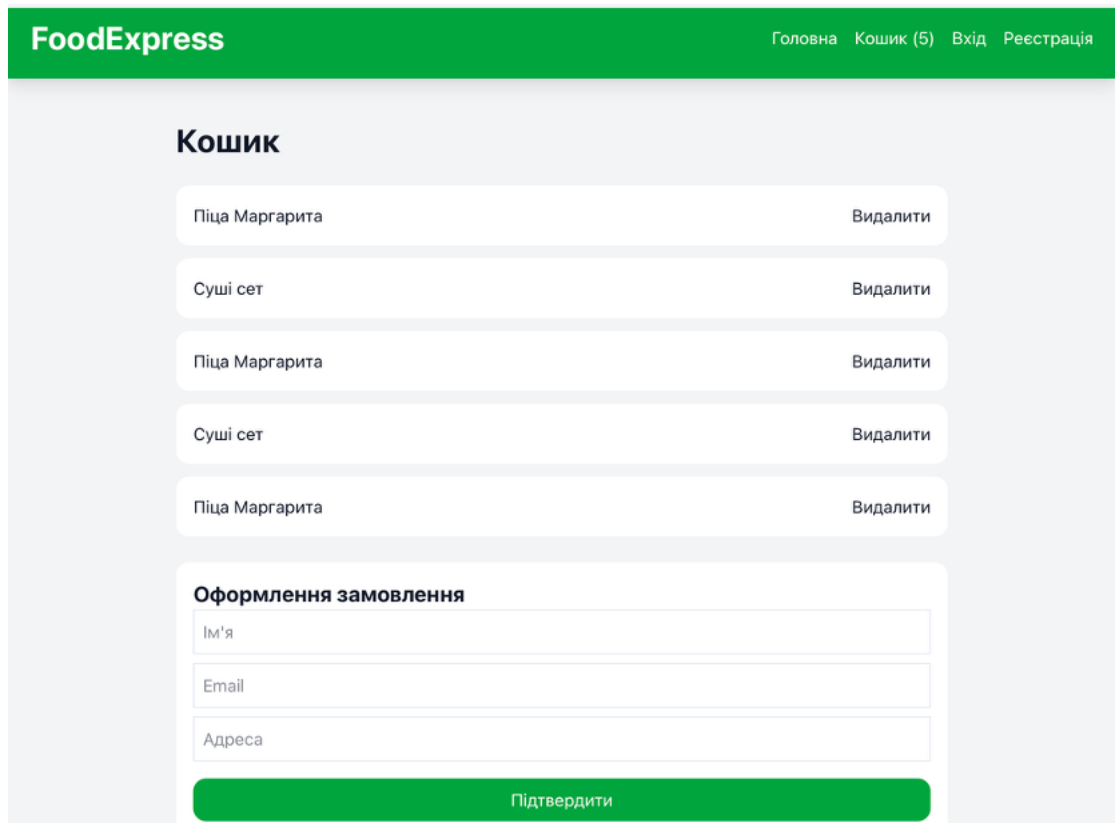
Рисунок 3.2 – Фільтрація товарів за категоріями

Відображення товарів реалізовано у вигляді сітки карток, кожна з яких містить зображення страви, її назву, ціну та кнопку додавання до кошика. Програмний код компонента відображення товарів наведено у лістингу 2.1.

Лістинг 2.1 – Фрагмент коду відображення товарів

```
{filteredProducts.map((p) => (
  <div key={p.id} className="bg-white rounded-2xl shadow p-4">
    <img src={p.image} className="rounded-xl h-40 w-full object-
cover" />
    <h3>{p.name}</h3>
    <p className="text-green-600 font-bold">{p.price}</p>
    <button onClick={() => addToCart(p)}>
      Додати
    </button>
  </div>
)})}
```

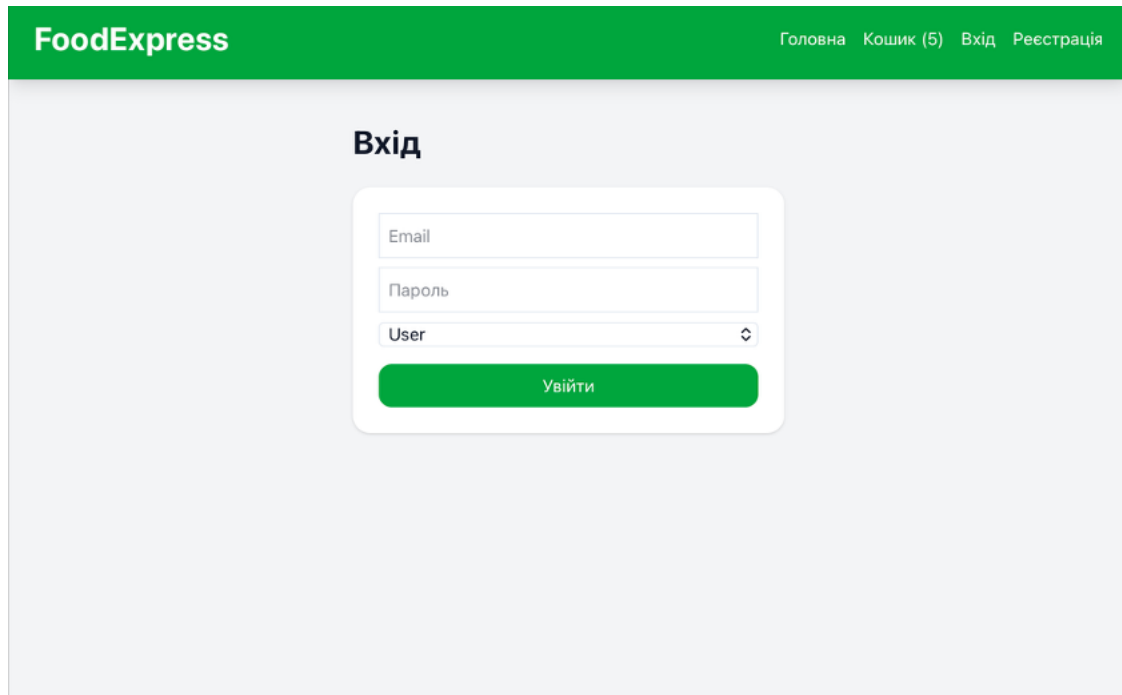
Наступним елементом інтерфейсу є кошик користувача. У кошику відображаються вибрані товари, передбачена можливість їх видалення, а також форма оформлення замовлення. Форма містить поля для введення імені, електронної пошти та адреси доставки зображено на рисунку 3.3



The screenshot shows the 'FoodExpress' website interface. At the top, there is a green navigation bar with the logo 'FoodExpress' on the left and links for 'Головна', 'Кошик (5)', 'Вхід', and 'Реєстрація' on the right. Below the navigation bar, the main content area is titled 'Кошик' (Cart). It displays a list of five items, each in a light gray rounded rectangle. The items are: 'Піца Маргарита' (Margherita Pizza), 'Суші сет' (Sushi Set), 'Піца Маргарита', 'Суші сет', and 'Піца Маргарита'. To the right of each item name is a 'Видалити' (Delete) button. Below the cart items is a section titled 'Оформлення замовлення' (Order Form). This section contains three input fields: 'Ім'я' (Name), 'Email', and 'Адреса' (Address). At the bottom of the form is a prominent green button labeled 'Підтвердити' (Confirm).

Рисунок 3.3 – Сторінка кошика та оформлення замовлення

Для забезпечення базового механізму автентифікації користувача реалізовано сторінку входу. Вона містить поля для введення електронної пошти та пароля, а також можливість вибору ролі користувача (звичайний користувач або адміністратор). Зовнішній вигляд форми входу наведено на рисунку 3.4.



FoodExpress

Головна Кошик (5) Вхід Реєстрація

Вхід

Email

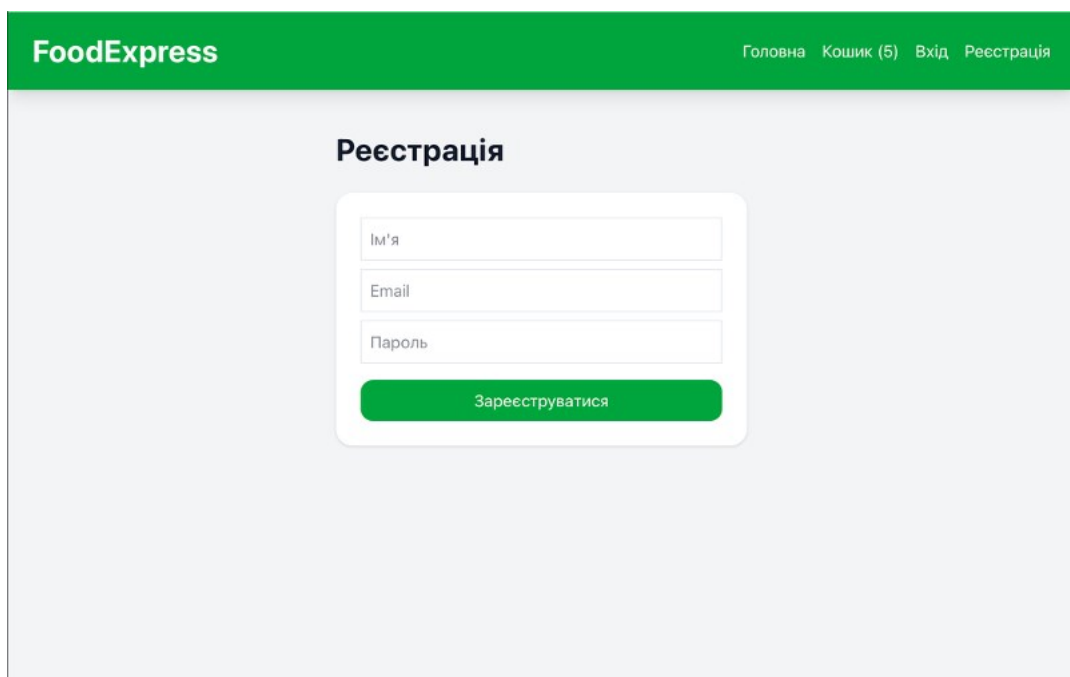
Пароль

User

Увійти

Рисунок 3.4 – Форма входу користувача

Також реалізовано сторінку реєстрації, яка дозволяє створити нового користувача шляхом введення основних даних: імені, електронної пошти та пароля. Дана форма забезпечує базовий рівень підготовки користувача до роботи із системою відображено на рисунку 3.5



FoodExpress

Головна Кошик (5) Вхід Реєстрація

Реєстрація

Ім'я

Email

Пароль

Зареєструватися

Рисунок 3.5 – Форма реєстрації користувача

Загалом, розроблений інтерфейс забезпечує логічну структуру навігації між сторінками, інтуїтивно зрозумілу взаємодію користувача з системою та базовий функціонал онлайн платформи замовлення їжі. Реалізоване рішення дозволяє у подальшому легко масштабувати функціональні можливості вебзастосунку, зокрема додавання повноцінної серверної частини та системи авторизації [11].

Після розробки інтерфейсу користувача було реалізовано основні функції для обробки даних вебплатформи онлайн замовлення та доставки харчових продуктів. Дані функції забезпечують взаємодію між окремими елементами інтерфейсу, обробку дій користувача та зміну стану застосунку.

Для реалізації логіки роботи застосунку використано механізм `useState` бібліотеки `React`, який дозволяє зберігати та оновлювати стан компонентів у режимі реального часу. За допомогою станів реалізовано зберігання інформації про товари, кошик, фільтрацію, дані користувача, а також навігацію між сторінками.

Однією з основних функцій є додавання товарів до кошика. При натисканні кнопки «Додати» вибраний товар передається до масиву `cart`, після чого автоматично оновлюється кількість товарів у кошику. Фрагмент програмного коду наведено у лістингу 2.2.

Лістинг 2.2 – Функція додавання товару до кошика

```
const addToCart = (product) => {  
  setCart([...cart, product]);  
};
```

Також реалізовано функцію видалення товарів із кошика. Вона дозволяє користувачу редагувати список вибраних товарів перед оформленням замовлення. Для цього використовується метод `filter`, який формує новий масив без вибраного елемента (лістинг 2.3).

Лістинг 2.3 – Функція видалення товару з кошика

```
const removeFromCart = (id) => {  
  setCart(cart.filter((p) => p.id !== id));  
};
```

Для покращення зручності користування вебплатформою реалізовано механізм фільтрації товарів за категоріями. Після вибору певної категорії система відображає лише ті товари, які належать до відповідного типу. Це дозволяє швидше знаходити необхідні страви серед великої кількості товарів.

Для реалізації переходів між сторінками застосунку використовується окремий стан `page`. Залежно від значення даної змінної відображається відповідний компонент інтерфейсу: головна сторінка, кошик, сторінка входу або реєстрації [12-13].

Наступною реалізованою функцією є авторизація користувача. Форма входу дозволяє вводити електронну пошту, пароль та обирати роль користувача (користувач або адміністратор). Після натискання кнопки входу інформація зберігається у стані `user`, що дозволяє відображати дані авторизованого користувача в інтерфейсі системи. Приклад функції входу наведено у лістингу 2.4.

Лістинг 2.4 – Функція авторизації користувача

```
const handleLogin = () => {
  if (!loginData.email || !loginData.password) return;

  setUser({
    email: loginData.email,
    role: loginData.role,
  });

  setPage("home");
};
```

Також реалізовано функцію виходу із системи, яка очищає інформацію про поточного користувача та повертає його на головну сторінку вебзастосунку.

Для оформлення замовлення створено форму введення даних користувача. Вона дозволяє зберігати введену інформацію у відповідному стані `form` та використовувати її при підтвердженні замовлення [14].

Загалом, реалізовані функції забезпечують базову логіку роботи вебплатформи, взаємодію користувача із системою та динамічне оновлення інтерфейсу без перезавантаження сторінки. Використаний підхід дозволяє легко розширювати функціональні можливості застосунку у подальшій розробці.

Повний програмний код реалізованих функцій та компонентів вебплатформи представлено у додатку А.

3.3 Тестування розробленого програмного комплексу

Розпочати тестування розробленого програмного комплексу доцільно з перевірки адміністративної панелі, оскільки саме вона забезпечує основний функціонал керування вебплатформою. Як показано на рисунку 3.6, головний інтерфейс адміністратора містить верхнє навігаційне меню та бічну панель керування, які забезпечують швидкий доступ до основних модулів системи.

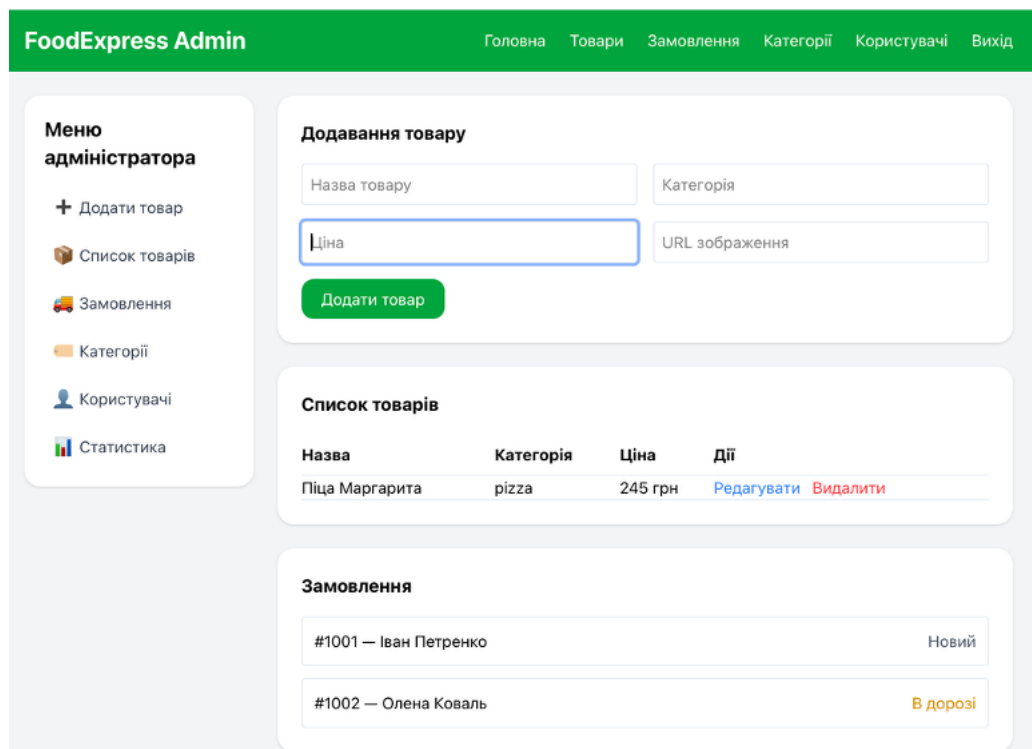


Рисунок 3.6 – Головний інтерфейс адміністративної панелі вебплатформи

Під час тестування було перевірено коректність відображення всіх елементів інтерфейсу, включаючи навігаційне меню, блоки контенту та адаптивність сторінки. Встановлено, що всі компоненти відображаються коректно та не перекривають один одного при зміні розміру вікна браузера.

Наступним етапом тестування є перевірка функціоналу додавання нових товарів до системи. На рисунку 3.7 зображено форму введення даних нового

товару, яка містить поля для введення назви, категорії, ціни та посилання на зображення.

Додавання товару

ТИТ

Категорія

оли

URL зображення

Додати товар

Рисунок 3.7 – Форма додавання нового товару

У ході тестування перевірено коректність введення даних у всі поля форми. Після натискання кнопки «Додати товар» інформація коректно відображається у списку товарів. Також перевірено сценарії введення некоректних або неповних даних, у результаті чого система не допускає додавання неповного запису.

Далі було протестовано модуль відображення списку товарів (рисунок 3.8). У даному блоці реалізовано табличне представлення інформації про товари, що дозволяє адміністратору швидко переглядати, редагувати та видаляти записи.

FoodExpress Admin Головна Товари Замовлення Категорії Користувачі Вихід

Табличне відображення товарів

ID	Назва	Категорія	Ціна	Дії
1	Піца Маргарита	pizza	245 грн	Редагувати Видалити
2	Суші сет	sushi	390 грн	Редагувати Видалити
3	Бургер	burger	180 грн	Редагувати Видалити

Рисунок 3.8 – Табличне відображення списку товарів

Під час тестування перевірено роботу функцій редагування та видалення товарів. Встановлено, що при видаленні запису він одразу зникає зі списку без помилок у відображенні інтерфейсу. Функція редагування забезпечує можливість зміни параметрів товару без порушення структури даних.

Окремо було протестовано модуль управління замовленнями клієнтів (рисунок 3.9). У даному розділі адміністратор має можливість переглядати список замовлень, а також змінювати їх статус залежно від етапу обробки.

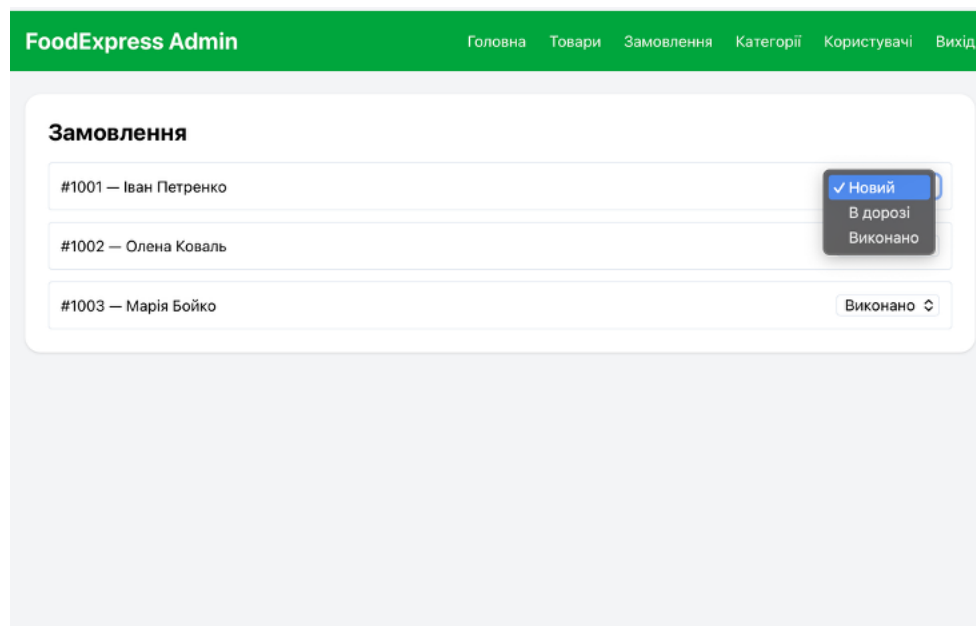


Рисунок 3.9 – Модуль перегляду та управління замовленнями

Під час тестування перевірено зміну статусів замовлень (наприклад: «Новий», «В дорозі», «Виконано»). Встановлено, що всі зміни одразу відображаються в інтерфейсі без перезавантаження сторінки.

Важливо зазначити, що інтерфейс користувацької частини системи було реалізовано та описано в пункті 3.4.2, де передбачено перегляд товарів, додавання їх до кошика та оформлення замовлення. Таким чином забезпечено розділення функціоналу між користувачем та адміністратором.[15]

У процесі тестування також було перевірено взаємодію між різними модулями системи. Зокрема, перевірено, що додані товари автоматично відображаються у користувацькій частині системи, а замовлення, створені користувачами, коректно відображаються в адміністративній панелі.

В результаті проведеного тестування встановлено, що програмний комплекс працює стабільно, усі основні функції виконуються коректно, помилки у роботі інтерфейсу не виявлені. Система відповідає вимогам, визначеним на етапі проєктування, та забезпечує повний цикл роботи вебплатформи для онлайн-замовлення

3.4 Перспективи модернізації

Даний програмний комплекс для онлайн-замовлення та доставки харчових продуктів виконує свої основні функції, однак може бути розширений та вдосконалений у подальшому розвитку. Модернізація системи дозволить автоматизувати додаткові процеси, підвищити продуктивність роботи та покращити зручність використання для користувачів і адміністраторів.

У перспективі доцільно реалізувати систему аналітики замовлень, яка дозволить відстежувати популярність товарів, формувати статистичні звіти та аналізувати ефективність роботи платформи. Це дозволить адміністраторам приймати більш обґрунтовані рішення щодо асортименту продукції.

Також важливим напрямком розвитку є впровадження рекомендаційної системи, яка буде пропонувати користувачам товари на основі їхніх попередніх замовлень або переглядів. Це підвищить рівень персоналізації сервісу та сприятиме збільшенню кількості замовлень.

Окремо можна реалізувати мобільну версію застосунку або повноцінний мобільний додаток, що забезпечить зручний доступ до сервісу з будь-якого пристрою.

Крім того, перспективним є розширення адміністративної панелі шляхом додавання автоматичного формування звітності та інтеграції з платіжними системами, що дозволить повністю автоматизувати процес обробки замовлень.

3.5 Висновок до другого розділу

У другому розділі було описано процес проектування та розробки програмного комплексу для онлайн-замовлення та доставки харчових продуктів. Було сформовано загальну структуру системи, визначено основні функціональні модулі та принципи їх взаємодії.

На етапі проектування бази даних було визначено основні сутності системи, серед яких товари, категорії, користувачі та замовлення, а також встановлено зв'язки між ними, що забезпечує цілісність та коректність збереження інформації.

Далі було розроблено інтерфейс користувача та адміністративну панель, які забезпечують зручну взаємодію з системою. Особливу увагу приділено структурі сторінок та основним елементам керування даними.

Також у розділі описано реалізацію функціональних можливостей системи, включаючи роботу з товарами та замовленнями, а також механізми зміни статусів і управління інформацією.[16-17]

Завершальним етапом стало тестування програмного комплексу, під час якого перевірено коректність роботи всіх основних модулів. Результати тестування показали стабільну роботу системи та відповідність її функціональних можливостей поставленим вимогам.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Заходи щодо автоматизації виробничих процесів, які сприяють покращенню умов праці

У сучасних умовах розвитку інформаційних технологій автоматизація виробничих процесів є одним із найважливіших напрямів підвищення ефективності діяльності підприємств та покращення умов праці працівників. Впровадження автоматизованих систем дозволяє зменшити обсяг ручної роботи, підвищити продуктивність праці, скоротити час виконання операцій та мінімізувати вплив людського фактора на результати роботи. Особливо актуальним це питання є для підприємств, діяльність яких пов'язана з обробкою великої кількості інформації та використанням комп'ютерної техніки.

Автоматизація виробничих процесів спрямована не лише на підвищення економічної ефективності підприємства, але й на створення безпечних та комфортних умов праці для персоналу. Використання сучасних програмних комплексів дозволяє значно зменшити фізичні та психологічні навантаження на працівників, знизити рівень стресу та підвищити якість виконання професійних обов'язків [18-19].

У межах розробленої вебплатформи для онлайн замовлення та доставки харчових продуктів автоматизація реалізується шляхом впровадження програмних засобів, які забезпечують автоматичне приймання, обробку та зберігання інформації про товари, користувачів та замовлення. Система дозволяє автоматично виконувати значну кількість операцій, які раніше потребували ручного введення або обробки даних [20].

Одним із важливих напрямів автоматизації є автоматизація процесу оформлення замовлень. Після вибору товарів користувачем система самостійно формує замовлення, обчислює загальну вартість покупки, зберігає інформацію у базі даних та передає її для подальшої обробки. Завдяки цьому працівники не витрачають додатковий час на ручне внесення інформації та перевірку правильності даних.

Суттєве значення має автоматизація управління асортиментом продукції. Адміністратор системи може додавати нові товари, редагувати їх характеристики, змінювати ціни та видаляти неактуальні позиції через зручний інтерфейс керування. Це значно спрощує процес адміністрування та зменшує ймовірність виникнення помилок під час роботи з інформацією.

Важливим заходом автоматизації є використання централізованої бази даних. Зберігання інформації в єдиній базі дозволяє забезпечити швидкий доступ до необхідних даних, виключити дублювання інформації та підвищити рівень контролю над процесами, що відбуваються в системі. Працівники можуть оперативно отримувати необхідну інформацію без пошуку документів або ручного аналізу великої кількості записів.

Значну роль у покращенні умов праці відіграє автоматизація процесу обробки та контролю замовлень. Система автоматично відображає статус кожного замовлення, що дозволяє адміністраторам контролювати процес доставки та своєчасно реагувати на можливі проблеми. Це зменшує навантаження на працівників та підвищує ефективність їх роботи.

Ще одним важливим напрямом є автоматизація формування звітної документації. Програмний комплекс дозволяє швидко отримувати інформацію про кількість замовлень, обсяги продажів, популярність товарів та інші показники діяльності. Автоматичне формування звітів значно скорочує час, необхідний для підготовки аналітичної інформації, та усуває помилки, які можуть виникати під час ручних розрахунків.

Автоматизація виробничих процесів також позитивно впливає на ергономічні умови праці. Завдяки використанню сучасного програмного забезпечення зменшується кількість одноманітних операцій, які виконуються працівниками протягом робочого дня. Це сприяє зниженню нервово-емоційного напруження, підвищенню концентрації уваги та зменшенню ризику професійного перевтомлення.

Крім того, автоматизовані системи забезпечують більш ефективну організацію робочого часу. Працівники можуть швидше виконувати свої обов'язки та приділяти більше уваги завданням, які потребують аналізу,

прийняття рішень або взаємодії з клієнтами. Це підвищує продуктивність праці та покращує загальну якість обслуговування користувачів.

Таким чином, впровадження заходів щодо автоматизації виробничих процесів є важливим фактором покращення умов праці. Автоматизація дозволяє скоротити обсяг ручної роботи, підвищити продуктивність персоналу, зменшити рівень психоемоційного навантаження та створити більш комфортні й безпечні умови праці. Використання сучасних програмних комплексів, зокрема вебплатформи для онлайн замовлення та доставки харчових продуктів, сприяє підвищенню ефективності роботи підприємства та забезпечує оптимізацію основних виробничих процесів.

4.2 Естетичне оформлення робочого місця оператора ПК, станка, установки

Естетичне оформлення робочого місця є важливим фактором забезпечення комфортних умов праці та підтримання високої працездатності працівників. Правильно організоване робоче середовище позитивно впливає на психоемоційний стан людини, сприяє зниженню рівня втоми, підвищенню концентрації уваги та ефективності виконання професійних обов'язків.

Для оператора персонального комп'ютера особливе значення має раціональне розміщення обладнання та елементів робочого місця. Робочий стіл повинен мати достатню площу для розташування монітора, клавіатури, миші та необхідної документації. Поверхня столу повинна бути матовою, щоб уникати появи відблисків та надмірного навантаження на органи зору.

Монітор рекомендується встановлювати на відстані 50–70 см від очей користувача. Верхня межа екрана повинна знаходитися на рівні очей або трохи нижче, що дозволяє підтримувати правильне положення голови та шиї під час роботи. Клавіатура та комп'ютерна миша повинні розташовуватися таким чином, щоб забезпечувати природне положення рук і мінімізувати навантаження на суглоби та м'язи.

Важливим елементом естетичного оформлення є кольорове оформлення приміщення. Для робочих кабінетів рекомендується використовувати світлі та спокійні кольори, такі як світло-бежевий, світло-зелений, світло-блакитний або світло-сірий. Дані кольори позитивно впливають на нервову систему людини, сприяють створенню комфортної атмосфери та зменшують психологічне навантаження під час тривалої роботи.

Освітлення робочого місця повинно забезпечувати комфортне сприйняття інформації на екрані монітора та паперових носіях. Перевагу слід надавати природному освітленню, а за його недостатності використовувати комбіноване штучне освітлення. Світлові прилади необхідно розташовувати таким чином, щоб уникати появи відблисків на поверхні монітора.

Для створення сприятливого психологічного середовища доцільним є використання елементів озеленення. Наявність кімнатних рослин покращує естетичний вигляд приміщення, сприяє очищенню повітря та позитивно впливає на емоційний стан працівників. Крім того, рослини допомагають зменшити рівень стресу та створюють більш комфортну атмосферу в робочій зоні.

Особливу увагу необхідно приділяти підтриманню порядку на робочому місці. Надмірна кількість сторонніх предметів та безлад можуть негативно впливати на концентрацію уваги та знижувати продуктивність праці. Робоча зона повинна містити лише ті предмети та документи, які необхідні для виконання поточних завдань.

Сучасні вимоги до організації робочого місця передбачають також використання ергономічних меблів. Робоче крісло повинно мати регульовану висоту сидіння, спинку та підлокітники. Це дозволяє забезпечити правильне положення тіла працівника та зменшити навантаження на хребет під час тривалої роботи за комп'ютером.

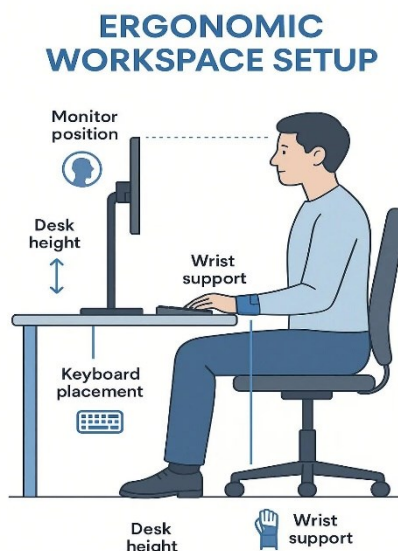


Рисунок 4.1 — Організація робочого місця оператора персонального комп'ютера

На рисунку 4.1 зображено приклад раціональної організації робочого місця оператора персонального комп'ютера. Правильне розташування обладнання, достатній рівень освітлення та використання ергономічних меблів сприяють зменшенню втоми працівника, підвищенню продуктивності праці та створенню комфортних умов роботи.

Для створення сприятливих умов праці важливе значення має також кольорове оформлення приміщення. Вибір кольорової гами повинен враховувати особливості трудової діяльності та психологічний вплив кольорів на людину зображено в таблиці 4.1.

Таблиця 4.1 — Рекомендовані кольори для оформлення робочих приміщень

Колір	Вплив на працівника
Світло-блакитний	Знижує втому та нервеве напруження
Світло-зелений	Сприяє концентрації уваги
Світло-сірий	Створює спокійну робочу атмосферу
Бежевий	Забезпечує психологічний комфорт
Білий	Візуально збільшує простір

Дані кольори рекомендуються для оформлення приміщень, у яких працівники тривалий час працюють за персональними комп'ютерами, оскільки вони позитивно впливають на працездатність, знижують рівень втоми та сприяють підтриманню комфортного психологічного стану.

Таким чином, естетичне оформлення робочого місця оператора ПК є важливим чинником забезпечення комфортних та безпечних умов праці. Раціональна організація робочого простору, правильне освітлення, використання ергономічних меблів, озеленення приміщення та гармонійне кольорове оформлення сприяють підвищенню працездатності працівників, зниженню рівня втоми та покращенню загального психологічного клімату в колективі.

4.3 Висновок до четвертого розділу

У четвертому розділі було розглянуто питання безпеки життєдіяльності та охорони праці під час роботи користувачів із розробленою вебплатформою для онлайн замовлення та доставки харчових продуктів. Особливу увагу приділено заходам щодо автоматизації виробничих процесів та організації робочого місця оператора персонального комп'ютера.

Проведений аналіз показав, що використання сучасних інформаційних технологій та програмних комплексів сприяє автоматизації значної кількості виробничих операцій, зменшенню обсягів ручної праці та підвищенню продуктивності роботи персоналу. Автоматизація процесів обробки замовлень, зберігання інформації та формування звітності дозволяє знизити рівень психоемоційного навантаження на працівників і мінімізувати ймовірність виникнення помилок під час виконання професійних обов'язків.

Також було досліджено основні вимоги до естетичного оформлення робочого місця оператора ПК. Встановлено, що раціональне розташування обладнання, належне освітлення, використання ергономічних меблів, підтримання порядку на робочому місці та створення сприятливого мікроклімату позитивно впливають на працездатність працівника, знижують рівень втоми та сприяють підвищенню ефективності праці.

Отже, дотримання вимог охорони праці, забезпечення безпечних умов роботи та впровадження заходів автоматизації є важливими складовими ефективного функціонування підприємства. Використання розробленої вебплатформи дозволяє не лише підвищити якість обслуговування користувачів, але й створити комфортні та безпечні умови праці для персоналу, що працює з інформаційною системою.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено вебплатформу для онлайн-замовлення та доставки харчових продуктів, яка забезпечує автоматизацію процесу оформлення замовлень, ефективну взаємодію між користувачами та адміністраторами, а також зручний доступ до каталогу продукції. Створений програмний комплекс дозволяє оптимізувати процес придбання товарів, підвищити якість обслуговування клієнтів та спростити управління замовленнями.

У першому розділі кваліфікаційної роботи було проведено аналіз предметної області, досліджено особливості функціонування сучасних вебплатформ для онлайн-доставки харчових продуктів та визначено основні вимоги до програмного комплексу. Також було проаналізовано існуючі аналоги, визначено акторів і сценарії використання системи, обґрунтовано вибір середовища розробки та сучасних програмних технологій, необхідних для реалізації вебплатформи.

У другому розділі було виконано проєктування програмного комплексу. Сформовано структуру вебплатформи, спроектовано її архітектуру та модель взаємодії основних компонентів, розроблено структуру бази даних і алгоритм функціонування системи. Запропоновані проєктні рішення забезпечують ефективний обмін інформацією між клієнтською та серверною частинами, високу продуктивність, масштабованість та можливість подальшого розширення функціональних можливостей вебплатформи.

У третьому розділі було реалізовано програмний комплекс відповідно до сформованих вимог. Розроблено користувацький інтерфейс та адміністративну панель, реалізовано функції реєстрації та авторизації користувачів, перегляду каталогу товарів, формування кошика, оформлення замовлень та їх обробки адміністраторами. Проведене тестування підтвердило коректність роботи всіх основних модулів системи, її стабільність та відповідність поставленим вимогам.

Також було визначено перспективи подальшої модернізації програмного комплексу.

У четвертому розділі розглянуто питання безпеки життєдіяльності та основ охорони праці під час роботи з персональним комп'ютером. Проаналізовано основні фактори, що впливають на працездатність користувача, охарактеризовано фази працездатності людини-оператора та наведено рекомендації щодо організації безпечного робочого місця відповідно до чинних вимог охорони праці.

Отже, поставлена мета кваліфікаційної роботи була повністю досягнута, а всі визначені завдання успішно виконані. Розроблена вебплатформа відповідає сучасним вимогам до інформаційних систем електронної комерції, забезпечує автоматизацію процесу онлайн-замовлення та доставки харчових продуктів, має зручний інтерфейс, високий рівень функціональності та значний потенціал для подальшого розвитку й масштабування.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Database, SQL Analytics, SQL Certification, SQL Data Analysis. Independently Published, 2019.
- 2 Довідник по CSS властивостям. Український веб-довідник. URL: <https://css.in.ua/css/properties>.
- 3 Довідник по HTML тегам. Український веб-довідник. URL: <https://css.in.ua/html/tags>.
- 4 Columns. Bootstrap The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.3/layout/columns/>.
- 5 Роббинс Дж. HTML5. Кишеньковий довідник / trans. by А. Сергеев. 5th ed. Науковий світ, 2022. 190 p.
- 6 Forms. Bootstrap The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.3/forms/overview/>.
- 7 Bootstrap icons. Bootstrap Icons · Official open source SVG icon library for Bootstrap. URL: <https://icons.getbootstrap.com/>.
- 8 HTML і CSS довідник українською. Html CSS довідник. URL: <https://html-css.co.ua/>.
- 9 Kumar R. Complete HTML Practical Course: HTML. Independently Published, 2020.
- 10 Grippa V. M., Kuzmichev S. Learning MySQL. O'Reilly Media, Incorporated, 2021.
- 11 MySQL Tutorial - Learn MySQL Fast, Easy and Fun. MySQL Tutorial. URL: <https://www.mysqltutorial.org/>.
- 12 Node.js. Node.js Official Documentation. URL: <https://nodejs.org/en/docs/>
- 13 Mongoose. Elegant MongoDB object modeling for Node.js. URL: <https://mongoosejs.com/docs/>
- 14 REST API Design Guidelines. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>
- 15 React. React – A JavaScript library for building user interfaces. URL: <https://react.dev/>

- 16 Бедрій Я. І. Безпека життєдіяльності : навч. посіб. Київ : Кондор, 2009. 286 с.
- 17 Працездатність людини-оператора. StudFiles. URL: <https://studfile.net/preview/4474247/page:11/>
- 18 Грибан В.Г., Негодченко О.В. Охорона праці. – К.: Центр учбової літератури, 2009ю 209 с.
- 19 ДСТУ 7234:2011. Дизайн і ергономіка. Обладнання виробниче. Загальні вимоги дизайну та ергономіки. Чинний від 2011-08-01. Вид. офіц. Укр. НДІ дизайну та ергономіки НАУ.
- 20 ДСТУ 7299:2013. Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки. Чинний від 2014-01-01. Вид. офіц.
- 21 Інструкція з охорони праці при роботі з комп'ютером, принтером, ксероксом та іншою оргтехнікою. Інструкції для навчальних закладів України. Інструкції для навчальних закладів України | Інструкції з охорони праці, техніки безпеки і пожежної безпеки. URL: <https://osvita-docs.com/node/41>
- 22 Leshchyshyn, Y., Scherbak, L., Nazarevych, O., Gotovych, V., Tymkiv, P., & Shymchuk, G. (2019, May). Multicomponent Model of the Heart Rate Variability Change-point. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) (pp. 110-113). IEEE.
- 23 Lytvynenko, I., Lupenko, S., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021, September). Mathematical model of gas consumption process in the form of cyclic random process. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 1, pp. 232-235). IEEE.
- 24 Kozlovskyi, V., Balanyuk, Y., Martyniuk, H., Nazarevych, O., Scherbak, L., & Shymchuk, G. (2022, April). Information Technology for Estimating City Gas Consumption During the Year. In 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan (pp. 1-4).
- 25 Lytvynenko, I., Lupenko, S., Kunanets, N., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021). Simulation of gas consumption process based on the mathematical model in the form of cyclic random process considering the scale factors.

In 1st International Workshop on Information Technologies: Theoretical and Applied Problems, ITTAP (Vol. 2021).

26 Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проектування архітектури програмних систем в проектах з гнучкими методами управління. Матеріали XXI наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 46-48.

27 Lupenko, S., Lytvynenko, I., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021, December). Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process. In Proceedings of the International Conference „Advanced applied energy and information technologies 2021”, 2021 (pp. 213-219). TNTU, Zhytomyr «Publishing house „Book-Druk “» LLC.

28 Lytvynenko, I., Lupenko, S., Nazarevych, O., Shymchuk, H., & Hotovych, V. (2022). Additive mathematical model of gas consumption process. Вісник Тернопільського національного технічного університету, 104(4), 87-97.

29 Kunanets, N., Pasichnyk, V., Bodnarchuk, I., Martsenko, S., Matsiuk, O., Matsiuk, A., ... & Shymchuk, H. (2019). Information system for visual analyzer disease diagnostics. In CEUR Workshop Proceedings (pp. 43-56).

30 Leschyshyn, Y. Z., Nazarevych, O. B., Shymchuk, G. V., Revutskyi, E. A., & Shcherbak, L. M. (2016, September). The Methods of Change Point Detection and Statistical Estimating of Dynamic of the Noise Stochastic Signals Characteristics. In THE SEVENTH WORLD CONGRESS “AVIATION IN THE XXI-st CENTURY” Safety in Aviation and Space Technologies September 19-21, NATIONAL AVIATION UNIVERSITY. Kyiv: NAU.

31 Nazarevych, O., Leshchyshyn, Y., Lupenko, S., Hotovych, V., Shymchuk, G., & Shabliy, N. (2020, September). Method of Gas Consumption Change-point Detection Based on Seasonally Multicomponent Model. In 2020 10th International Conference on Advanced Computer Information Technologies (ACIT) (pp. 152-155). IEEE.

32 ШИМЧУК, Г., ШЕВЧЕНКО, Н., ШВИРЛО, К., & ГАРМАТЮК, Н. (2025). СИСТЕМА ВІДНОВЛЕННЯ ДАНИХ У БЕЗДРОТОВИХ СЕНСОРНИХ

МЕРЕЖАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical sciences, 353(3.2), 246-250.

33 Шимчук, Г., Голотенко, О., & Золотий, Р. З. (2022). Основні проблеми та загрози хмарної безпеки. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 59-60.

34 Шимчук, Г. В., Маєвський, О. В., & Назаревич, О. Б. (2016). Конспект лекцій з дисципліни «Розподілені системи моніторингу та керування».

35 Palianytsia, Y., Lytvynenko, I., Menoub, A., Shymchuk, H., & Dubchak, A. (2024). Development of an algorithm for identification of damage types on the surface of sheet metal.

36 Shymchuk, G., Lytvynenko, I., Hromyak, R., Lytvynenko, S., & Hotovych, V. (2023). Gas Consumption Forecasting Using Machine Learning Methods and Taking Into Account Climatic Indicators. In CITI (pp. 156-163).

37 Шимчук, Г. В., Маєвський, О. В., & Назаревич, О. Б. (2016). Конспект лекцій з дисципліни Комп'ютерна графіка для студентів освітнього рівня «бакалавр» спеціальності 125 «Кібербезпека».

38 Yasniy, O., Didych, I., Tymoshchuk, D., Pasternak, I., Nykytyuk, V., Shymchuk, H., & Radyk, D. (2026). Fatigue crack growth prediction of automotive steels using ensemble-based machine learning methods. Procedia Structural Integrity, 81, 116-122.

39 Palka, O., Stanko, A., Shymchuk, H., & Herasymchuk, O. (2021). Запобігання поширення коронавірусної інфекції у «розумних містах». COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION, (42), 79-88.

40 Шимчук, Г. В., Назаревич, О. Б., Литвиненко, Я. В., Готович, В. А., Никитюк, В. В., & Боднарчук, І. О. (2025). Грід-системи та технології хмарних обчислень. Навчальний посібник для здобувачів освітнього рівня «магістр» спеціальностей: F3 «Комп'ютерні науки», F6 «Інформаційні системи та технології».

41 Sorokivskiy, O., Hotovych, V., Nazarevych, O., & Shymchuk, G. (2025). Comparative analysis of camera calibration algorithms for football applications. *Journal of Computer Vision in Sports*.

42 Шевченко, Н. А., Шимчук, Г. В., & Гарматюк, У. А. (2024). Оптимізація метрик маршрутизації для забезпечення стійкості та надійності IP-мереж. Збірник тез доповідей XIII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 400-401.

43 Шевченко, Н. А., Шимчук, Г. В., & Гарматюк, У. А. (2024). Інтеграція технології мережевої віртуалізації VRF у багатоколіїну маршрутизацію. Збірник тез доповідей XIII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 398-399.

44 Шимчук, Г. В. (2022). Дослідження методів захисту відомих хмарних платформ (Master's thesis, ТНТУ).

45 Шимчук Г., Голотенко О., Небесний Р., Готович В. Застосування мови Scala у системах паралельних і хмарних обчислень. *Наука і техніка сьогодні*. 2026. № 4(58). С. 4794–4807. DOI: 10.52058/2786-6025-2026-4(58)-4794-4807.

46 Шевченко Н., Шимчук Г., Готович В., Голотенко О., Литвиненко С., Петрошук М. Математична модель для прогнозування змін у бездротових сенсорних мережах. *Наука і техніка сьогодні*. 2026. № 4(58). С. 4767–4782. DOI: 10.52058/2786-6025-2026-4(58)-4767-4782.

ДОДАТКИ

Програмний код головної сторінки

```

<table class="table-bordered" style="font-family: Cambria; font-
size: 14pt; width: <?php echo 69+count($array_groups)*180; ?>px;">
  <tr>
    <td colspan="2" rowspan="3"></td>
    <?php for ($i = 0; $i < count($array_groups);$i++) { ?>
      <td colspan="4" style="font-size: 22pt; background-color:
#C0C0C0; text-align: center; font-weight: bold; width:
180px;"><?php echo $array_groups[$i]['name'] ?></td>
      <?php } ?>
    </tr>
    <tr>
      <?php for ($i = 0; $i < count($array_groups);$i++) { ?>
        <td colspan="4" style="font-style: italic; text-align:
center; width: 180px;" ><?php echo $array_groups[$i]['kurator']
?></td>
        <?php } ?>
      </tr>
      <tr>
        <?php for ($i = 0; $i < count($array_groups); $i++) {
          if (!empty($result['spec1'])) { ?>
            <td colspan="4" style="font-style: italic; text-align:
center; width: 180px;" ><?php echo $array_groups[$i]['spec1']
?></td>
            <?php } else { ?>
              <td colspan="2" class=" td-2" style="font-style: italic;
text-align: center; width: 90px; height: 22px;"><?php echo
$array_groups[$i]['spec1'] ?></td>
              <td colspan="2" class=" td-2" style="font-style: italic;
text-align: center; width: 90px; height: 22px;"><?php echo
$array_groups[$i]['spec2'] ?></td>
              <?php }
            } ?>
          </tr>
          <?php
          for ($i = 0; $i < count($array_day); $i++) {
            for ($j = 1; $j < 8; $j++) {
              if (!empty($_POST['teacher'])) {
                $sql="SELECT `groups`.`id` AS
`group`, `groups`.`id_department`, `schedule`.* FROM `schedule`
INNER JOIN `subjects` ON `schedule`.`id_subject`=`subjects`.`id`
INNER JOIN `groups` ON `subjects`.`id_group`=`groups`.`id` WHERE
`schedule`.`day`='".$i."' AND `schedule`.`n_lesson`='".$j."' AND (
";
                $sql=$sql."`schedule`.`id_subject`='".$array_s_id[0]."' ";
                for($idd=1;$idd<count($array_s_id);$idd++){
                  $sql=$sql."OR `schedule`.`id_subject`='".$array_s_id[$idd]."'
";
                }
                $sql=$sql.") ORDER BY `groups`.`id_department` ASC;";
              } else {

```

```

        $sql="SELECT `groups`.`id` AS `group`,
`groups`.`id_department`, `schedule`.* FROM `schedule` INNER JOIN
`subjects` ON `schedule`.`id_subject`=`subjects`.`id` INNER JOIN
`groups` ON `subjects`.`id_group`=`groups`.`id` WHERE
`schedule`.`day`='".$i."' AND `schedule`.`n_lesson`='".$j."' ORDER
BY `groups`.`id_department` ASC;";
    }
    $res=mysqli_query($connect,$sql);
    $array_1=[];
    while ($result = mysqli_fetch_array($res)) {
        $array_1[]=$result;
    }?>
    <tr>
        <?php if($j==1){ ?><td rowspan="28" style="font-size: 22pt;
font-weight: bold; width: 44px; text-align: center;"><?php echo
$array_day[$i][0]; ?></td><?php } ?>
        <td rowspan="4" style="font-weight: bold; width: 25px; text-
align: center;"><?php echo $j; ?></td>
        <?php
        for ($g = 0; $g < count($array_groups); $g++) {
            $n=0;
            for ($ar = 0; $ar < count($array_1); $ar++) {
                if($array_groups[$g]['id']==$array_1[$ar]['group']){
                    if($array_1[$ar]['id_type_lesson']=='1'){
                        echo "<td colspan='4' style='height: 23px; border-bottom:
hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]['id_cl
assroom1'], $array_cl)."</td>";
                        $n=1;
                    }
                    if($array_1[$ar]['id_type_lesson']=='2'){
                        echo "<td colspan='2' style='height: 23px; border-right:
hidden; border-bottom: hidden; text-align: left; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]['id_cl
assroom1'], $array_cl)."</td><td colspan='2' style='height: 23px;
border-bottom: hidden; text-align: right; font-style:
italic;'>".$array_1[$ar]['name_teacher']."</td>";
                        $n=1;
                    }
                    if($array_1[$ar]['id_type_lesson']=='3'){echo "";$n=1;}
                    if($array_1[$ar]['id_type_lesson']=='4'){
                        echo "<td colspan='2' style='height: 23px; border-bottom:
hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]['id_cl
assroom1'], $array_cl)."</td>";
                        $n=1;
                    }
                    if($array_1[$ar]['id_type_lesson']=='5'){
                        echo "<td colspan='2' style='height: 23px; border-bottom:
hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]['id_cl
assroom1'], $array_cl)."</td>";
                        $n=1;
                    }
                }
            }
        }
    }

```



```

        if($array_1[$ar]['id_type_lesson']=='6'){
            echo "<td colspan='2' style='height: 24px; text-align:
center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
        }
        if($array_1[$ar]['id_type_lesson']=='7'){echo "";}
        if($array_1[$ar]['id_type_lesson']=='8'){
            echo "<td colspan='2' style='height: 24px; text-align:
center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
        }
        if($array_1[$ar]['id_type_lesson']=='9'){echo ""; }
    }
}
}
?>
</tr>
<tr>
<?php
for ($g = 0; $g < count($array_groups); $g++) {
for ($ar = 0; $ar < count($array_1); $ar++) {
if($array_groups[$g]['id']==$array_1[$ar]['group']){
if($array_1[$ar]['id_type_lesson']=='1'){
echo "";
}
if($array_1[$ar]['id_type_lesson']=='2'){echo ""; }
if($array_1[$ar]['id_type_lesson']=='3'){
echo "<td colspan='4' style='height: 24px; border-bottom:
hidden; text-align: center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
}
if($array_1[$ar]['id_type_lesson']=='4'){echo "";}
if($array_1[$ar]['id_type_lesson']=='5'){echo "";}
if($array_1[$ar]['id_type_lesson']=='6'){echo "";}
if($array_1[$ar]['id_type_lesson']=='7'){
echo "<td colspan='2' style='height: 24px; border-bottom:
hidden; text-align: center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
}
if($array_1[$ar]['id_type_lesson']=='8'){echo ""; }
if($array_1[$ar]['id_type_lesson']=='9'){
echo "<td colspan='2' style='height: 24px; border-bottom:
hidden; text-align: center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
}
}
}
}
?>
</tr>
<tr>
<?php
for ($g = 0; $g < count($array_groups); $g++) {
for ($ar = 0; $ar < count($array_1); $ar++) {

```

```

        if($array_groups[$g]['id']==$array_1[$sar]['group']){
            if($array_1[$sar]['id_type_lesson']=='1'){
                echo "<td colspan='4' style='height: 23px; text-align: right;
font-style: italic;'>".$array_1[$sar]['name_teacher']."</td>";
            }
            if($array_1[$sar]['id_type_lesson']=='2'){echo "";}
            if($array_1[$sar]['id_type_lesson']=='3'){
                echo "<td colspan='2' style='height: 23px; border-right:
hidden; text-align: left; font-style: italic;
'>".clas($array_1[$sar]['id_classroom'],$array_1[$sar]['id_classroom
1'],$array_cl)."</td><td colspan='2' style='height: 23px; text-
align: right; font-style:
italic;'>".$array_1[$sar]['name_teacher']."</td>";
            }
            if($array_1[$sar]['id_type_lesson']=='4'){
                echo "<td colspan='2' style='height: 23px; text-align: right;
font-style: italic; ' >".$array_1[$sar]['name_teacher']."</td>";
            }
            if($array_1[$sar]['id_type_lesson']=='5'){
                echo "<td colspan='2' style='height: 23px; text-align: right;
font-style: italic; ' >".$array_1[$sar]['name_teacher']."</td>";
            }
            if($array_1[$sar]['id_type_lesson']=='6'){echo "";}
            if($array_1[$sar]['id_type_lesson']=='7'){
                echo "<td style='height: 23px; border-right: hidden; text-
align: left; font-style: italic;
'>".clas($array_1[$sar]['id_classroom'],$array_1[$sar]['id_classroom
1'],$array_cl)."</td><td style='height: 23px; text-align: right;
font-style: italic;'>".$array_1[$sar]['name_teacher']."</td>";
            }
            if($array_1[$sar]['id_type_lesson']=='8'){echo "";}
            if($array_1[$sar]['id_type_lesson']=='9'){
                echo "<td style='height: 23px; border-right: hidden; text-
align: left; font-style: italic;
'>".clas($array_1[$sar]['id_classroom'],$array_1[$sar]['id_classroom
1'],$array_cl)."</td><td style='height: 23px; text-align: right;
font-style: italic;'>".$array_1[$sar]['name_teacher']."</td>";
            }
        }
    }
}
?>
</tr>
<?php } }
</table>

```

Блок форми додавання нового продукту

```

echo '<div class="row"><div class="col-sm-11"><select
name="subj'.$name.'" id="subj'.$name.'"
onchange="updateInputValue'.$name.'" class="form-select">';
    $sql123="SELECT * FROM `subjects` WHERE `name`=' ' AND
`id_group`='".$array_groups['id'].'"";
    $res123=mysqli_query($connect,$sql123);
    $result123 = mysqli_fetch_array($res123);
    echo "<option value='".$result123['id']."'></option>";
    $sqlq="SELECT `schedule`.`id`, `subjects`.`id_teacher` FROM
`schedule` INNER JOIN `subjects` ON
`schedule`.`id_subject`=`subjects`.`id` WHERE
`schedule`.`day`='".$i.'" AND `schedule`.`n_lesson`='".$j.'" AND
`subjects`.`id_teacher`!=1' AND `subjects`.`name`!= ' ' AND
`schedule`.`zvedena`=0' UNION SELECT `schedule`.`id`,
`subjects`.`teachers` FROM `schedule` INNER JOIN `subjects` ON
`schedule`.`id_subject`=`subjects`.`id` WHERE
`schedule`.`day`='".$i.'" AND `schedule`.`n_lesson`='".$j.'" AND
`subjects`.`teachers`!=1' AND `subjects`.`name`!= ' ' ";
    $resq=mysqli_query($connect,$sqlq);
    $teach="";
    while ($resultq = mysqli_fetch_array($resq)) {
        $teach=$teach." AND
`id_teacher`!= '".$resultq['id_teacher'].'" AND
`teachers`!= '".$resultq['id_teacher'].'" ";
    }
    $sql="SELECT * FROM `subjects` WHERE `name`!= ' ' AND
`id_group`='".$array_groups['id'].'"'.$teach." ORDER BY `name`
ASC";
    $res=mysqli_query($connect,$sql);
    while ($result = mysqli_fetch_array($res)) {
        echo "<option value='".$result['id']."'>".$result['name'].
(".pib($result['id_teacher'], $result['teachers'],
$array_teachers_f).)".</option>";
    }
    $sql2="SELECT `id_department` FROM `groups` WHERE
`id`='".$g.'"";
    $res2=mysqli_query($connect,$sql2);
    $v = mysqli_fetch_array($res2);
    echo '</select></div><div class="col-sm-1"><a
href="http://r.gi.edu.ua/settings_subj.php?v='.$v[0].'#g'.$g.'"
target="_blank"><svg xmlns="http://www.w3.org/2000/svg" width="19"
height="19" fill="currentColor" class="bi bi-plus-circle"
viewBox="0 0 16 16"><path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0
1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/><path d="M8 4a.5.5 0 0 1
.5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5 0 0 1-1 0v-3h-3a.5.5 0 0 1 0-
1h3v-3A.5.5 0 0 1 8 4z"/></svg></a></div></div><br>';
    echo '<script>function updateInputValue'.$name.'"() { var
select = document.getElementById("subj'.$name.'"); var input =
document.getElementById("name_subj'.$name.'"); var input1 =
document.getElementById("name_teach'.$name.'"); switch(select.valu
e) {';

```

```

        $sql="SELECT * FROM `subjects` WHERE `name`!=' ' AND
`id_group`='". $array_groups['id']. "'          $res=mysqli_query($con
nect,$sql);
        while ($result = mysqli_fetch_array($res)) {
            echo " case '". $result['id']. "' :
input.value='". $result['name']. "' ;
input1.value='". pib($result['id_teacher'], $result['teachers'],
$array_teachers_f). "' ; break;";
        }
        echo 'default: input.value=" "; input1.value=" "; break; }
    }</script>';
    echo "<label class='form-label'>Відображається в
розкладі:</label><input name='name_subj'.". $name. "'
id='name_subj'.". $name. "' value='' type='text' class='form-
control'><br><input name='name_teach'.". $name. "'
id='name_teach'.". $name. "' value='' type='text' class='form-
control'><br>";
    echo '<select name="room'.'. $name. '[]" multiple class="form-
select" style="height: 20%;">';
    for ($z = 0; $z < count($array_classroom); $z++) {
        if($array_classroom[$z]['id']=='0'){
            echo "<option selected value='' . $array_classroom[$z]['id']
. '>' . $array_classroom[$z]['name'] . "</option>";
        } else {
            echo "<option value='' . $array_classroom[$z]['id'] . '>' .
$array_classroom[$z]['name'] . "
('.'. $array_classroom[$z]['nameK']. ".) ". "</option>";
        }
    }
    echo '</select><p>*Для вибору декількох кабінетів зажмите
Ctrl</p><input name="zvedena'.'. $name. ' "' type="checkbox"
value="1">Зведена пара<br><br>';

```

Код сторінки для редагування статусу

```

<div class="row">
  <div class="col-sm-1"></div>
  <div class="col-sm-2">
    <ul class="nav flex-column" style="margin-top: 15%;">
      <?php for ($i = 0; $i < count($array_department); $i++) {
        echo "<li class='nav-item'><a class='nav-link'
href='#v'.".$array_department[$i][0]."' style='color: #7e868d;
margin-bottom: 5%;'>".$array_department[$i][1]."</a></li>";
      } ?>
    </ul>
  </div>
  <div class="col-sm-6">
    <?php for ($i = 0; $i < count($array_department); $i++) { ?>
    <table class="table">
      <thead>
        <tr><th colspan="5" id="v<?php echo $array_department[$i][0];
?>"><?php echo $array_department[$i][1]; ?></th></tr>
        <tr>
          <th scope="col">№</th>
          <th scope="col">Назва групи</th>
          <th scope="col">Куратор</th>
          <th scope="col">Спеціальності</th>
          <th scope="col"></th>
        </tr>
      </thead>
      <tbody>
        <?php
          $sql="SELECT `groups`.*, `department`.`name` AS `d_name` FROM
`groups` INNER JOIN `department` ON
`groups`.`id_department`=`department`.`id` WHERE
`groups`.`id_department`='". $array_department[$i][0]."'";
          $res=mysqli_query($connect,$sql);
          $n=1;
          while($result=mysqli_fetch_array($res)) {
            <tr>
              <th scope="row"><?php echo $n; ?></th>
              <td><?php echo $result['name'] ?></td>
              <td><?php echo $result['kurator'] ?></td>
              <td><?php if (!empty($result['spec1'])) {
                echo $result['spec1'] . " | " . $result['spec2'];
              } ?></td>
              <td data-bs-toggle="modal" data-bs-
target="#staticBackdrop<?php echo $result['id'] ?>">
                <svg xmlns="http://www.w3.org/2000/svg" width="16"
height="16" fill="currentColor" class="bi bi-pencil " viewBox="0 0
16 16">
                  <path d="M12.146 14.6a.5.5 0 0 1 .708 0l3-.5.5 0 0 1 0
.708l-10 10a.5.5 0 0 1-.168 1.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0
0 1 .11-.168l10-10zm11.207 2.5l3.5 4.793 3.5 12.5 1.207
11.207 2.5zm1.586 3L10.5 3.207 4 9.707V10h.5a.5.5 0 0 1
.5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293l6.5-6.5zm-9.761 5.175-.106.106-

```

```

1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0
1-.5-.5V11h-.5a.5.5 0 0 1-.468-.325z"/>
</svg>
</td>
</tr>
<div class="modal fade" id="staticBackdrop<?php echo
$result['id'] ?>" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="staticBackdropLabel">
Редарувати</h5>
<button type="button" class="btn-close text-center" data-bs-
dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<form method="POST" action="settings_group.php">
<div class="form-floating mb-3">
<input type="hidden" name="id" value="<?php echo
$result['id'] ?>">
<input type="text" name="name" value="<?php echo
$result['name'] ?>" class="form-control" id="floatingInput"
placeholder="Назва групи">
<label for="floatingInput">Назва групи</label>
</div>
<div class="mb-3">
<label class="text-muted text-start" style="margin-bottom:
3px;">Відділення</label>
<select class="form-select" style="height: 55px;" name="vid">
<?php
for ($j= 0; $j < count($array_department); $j++){
echo "<option value='". $array_department[$j][0]. "' ";
if($array_department[$j][0]==$result['id_department']){
echo "selected";
}
echo ">". $array_department[$j][1]. "</option>";
}
?>
</select>
</div>
<div class="form-floating mb-3">
<input type="text" name="kurator" value="<?php echo
$result['kurator'] ?>" class="form-control" id="floatingInput"
placeholder="Куратор">
<label for="floatingInput">Куратор</label>
</div>
<hr>
<div class="form-floating mb-3">
<input type="text" name="spec1" value="<?php echo
$result['spec1'] ?>" class="form-control" id="floatingInput"
placeholder="Спеціальність 1">
<label for="floatingInput">Спеціальність 1</label>

```

```

</div>
<div class="form-floating mb-3">
  <input type="text" name="spec2" value="<?php echo
$result['spec2'] ?>" class="form-control" id="floatingInput"
placeholder="Спеціальність 2">
  <label for="floatingInput">Спеціальність 2</label>
</div>
<br>
<div class="form-floating mb-3 text-center">
  <button type="submit" name="save" class="btn btn-outline-
secondary" style="margin-right: 40%;">Зберегти</button>
  <button type="submit" name="delete" class="btn btn-outline-
secondary">Видалити</button>
</div>
</form>
</div>
<div class="modal-footer">
</div>
</div>
</div>
</div>
<?php $n++;} ?>
<tr>
  <td colspan="5" style="text-align: center;">
    <button type="button" data-bs-toggle="modal" data-bs-
target="#staticBackdropNew<?php echo
$array_department[$i][0]; ?>">
      <svg xmlns="http://www.w3.org/2000/svg" width="25"
height="25" fill="currentColor" class="bi bi-plus-circle"
viewBox="0 0 16 16">
        <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8
0a8 8 0 0 0 0 16z"/>
        <path d="M8 4a.5.5 0 0 1 .5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5 0
0 1-1 0v-3h-3a.5.5 0 0 1 0-1h3v-3A.5.5 0 0 1 8 4z"/>
      </svg>
    </button>
    <div class="modal fade" id="staticBackdropNew<?php echo
$array_department[$i][0]; ?>" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
      <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title" id="staticBackdropLabel">Додати
групу</h5>
            <button type="button" class="btn-close text-center" data-bs-
dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form method="POST" action="settings_group.php">
              <div class="form-floating mb-3">
                <input type="text" name="name" class="form-control"
id="floatingInput" placeholder="Назва групи">
                <label for="floatingInput">Назва групи</label>

```

```

</div>
<div class="mb-3">
<label class="text-muted text-start" style="margin-bottom:
3px;">Відділення</label>
<select class="form-select" style="height: 55px;" name="vid">
<?php
for ($q = 0; $q < count($array_department);$q++){
echo "<option value='". $array_department[$q][0]."' ";
if($array_department[$q][0]==$array_department[$i][0]){
echo "selected";
}
echo ">". $array_department[$q][1]."</option>";
} ?>
</select>
</div>
<div class="form-floating mb-3">
<input type="text" name="kurator" class="form-control"
id="floatingInput" placeholder="Куратор">
<label for="floatingInput">Куратор</label>
</div>
<hr><div class="form-floating mb-3">
<input type="text" name="spec1" class="form-control"
id="floatingInput" placeholder="Спеціальність 1">
<label for="floatingInput">Спеціальність 1</label>
</div>
<div class="form-floating mb-3">
<input type="text" name="spec2" class="form-control"
id="floatingInput" placeholder="Спеціальність 2">
<label for="floatingInput">Спеціальність 2</label>
</div>
<br><div class="form-floating text-center">
<button type="submit" name="new" class="btn btn-outline-
secondary">Зберегти</button>
</div>
</form>
</div>
<div class="modal-footer"></div>
</div>
</div>
</div>
</td>
</tr>
</tbody>
</table>
<br><?php } ?>
</div>
<div class="col-sm-1"></div>
</div>

```